Received June 03, 2018, accepted July 20, 2018, date of publication xxxx 00, 2018, date of current version July 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.DOI

Fog-supported delay-constrained energy-saving live migration of VMs over MultiPath TCP/IP 5G connections

ENZO BACCARELLI, MICHELE SCARPINITI, (SENIOR MEMBER, IEEE), AND ALIREZA MOMENZADEH

Department of Information Engineering, Electronics and Telecommunications (DIET), "Sapienza" University of Rome, Via Eudossiana 18, Rome 00184, Italy

Corresponding author: Michele Scarpiniti (michele.scarpiniti@uniroma1.it)

This work has been supported by the project: "GAUChO – A Green Adaptive Fog Computing and networking Architectures" funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 – grant 2015YPXH4W_004, and by the projects V-FOG and V-FOG2: "Vehicular Fog energy-efficient QoS mining and dissemination of multimedia Big Data streams" funded by Sapienza University of Rome, Bando 2016 and 2017.

ABSTRACT The incoming era of the Fifth-Generation Fog Computing-supported Radio Access Networks (shortly, 5G FOGRANs) aims at exploiting computing/networking resource virtualization, in order to augment the limited resources of wireless devices through the seamless live migration of Virtual Machines (VMs) towards nearby Fog data centers. For this purpose, the bandwidths of the multiple Wireless Network Interface Cards (WNICs) of the wireless devices may be aggregated under the control of the emerging MultiPathTCP (MPTCP) protocol. However, due to fading and mobility-induced phenomena, the energy consumptions of current state-of-the-art VM migration techniques may still offset their expected benefits. Motivated by these considerations, in this paper, we analytically characterize, implement in software and numerically test the optimal minimum-energy Settable-Complexity Bandwidth Manager (SCBM) for the live migration of VMs over 5G FOGRAN MPTCP connections. The key features of the proposed SCBM are that: (i) its implementation complexity is settable on-line on the basis of the target energy consumption-vs.-implementation complexity tradeoff; (ii) it minimizes the network energy consumed by the wireless device for sustaining the migration process under hard constraints on the tolerated migration times and downtimes; and, (iii) by leveraging a suitably designed adaptive mechanism, it is capable to quickly react to (possibly, unpredicted) fading and/or mobility-induced abrupt changes of the wireless environment without requiring forecasting. The actual effectiveness of the proposed SCBM is supported by extensive energy-vs.-delay performance comparisons, that cover: (i) a number of heterogeneous 3G/4G/WiFi FOGRAN scenarios; (ii) synthetic and real-world workloads; and, (iii) MPTCP and SinglePathTCP (SPTCP) wireless connections.

INDEX TERMS Live VM migration, 5G FOGRAN, Settable-Complexity Bandwidth Manager (SCBM), MPTCP, Energy efficiency, Delay-sensitive applications, Hard QoS constraints.

I. INTRODUCTION

Smartphones are already our ubiquitous technological assistants. Since 2011, the worldwide smartphone penetration overtook that of wired PCs by reaching 80% in US and Europe. Cisco envisions that the average number of connected mobile devices per person will reach 6.6 in 2020, mainly fostered by the expected pervasive use of Internet of Everything (IoE) applications [1]. This will be also due to the fact that future smartphones will increase their capability to use in an efficient way the multiple heterogeneous Wireless Network Interface Cards (WNICs) that already equip them by exploiting an emerging bandwidth-pooling transport technology, generally referred to as MultiPath TCP (MPTCP) [2].

However, since smartphones are (and will continue to be) resource and energy limited, Mobile Cloud Computing (MCC) is expected to effectively support them by providing device augmentation [3]. MCC is, indeed, a quite novel technology that relies on the synergic cooperation of three different paradigms, namely, cloud computing, mobile computing and wireless Internet. Its main goal is to cope with the inherent resource limitation of the mobile devices by allowing them to offload computation and/or memory-intensive applications (such as, for example, image processing, voice recognition, online gaming and social networks) towards virtualized data centers [3]. However, performing application offloading from mobile devices to remote large-scale cloud-based data centers (e.g., Amazon and Google, to name a few) suffers from the large communication delay and limited bandwidth typically offered by multi-hop cellular Wide Area Networks (WANs) [3]. The dream is, indeed, to allow mobile devices to: (i) exploit their multiple WNICs to perform bandwidth aggregation through *MPTCP* [2]; and, then: (ii) leverage the sub-millisecond access latencies promised by the emerging *Fifth Generation* (5G) multi Radio Access Networks (RANs) [4], in order to perform *seamless* application offloading towards *proximate* virtualized data centers, generally referred to as *Fog nodes* [1]. For this purpose, the application to be offloaded from the mobile device is shipped in the form of a *Virtual Machine* (VM) and, then, the offloading process takes place as a *live* (e.g., seamless) VM *migration* from the mobile device to the serving Fog node [5].

A. CONVERGENCE OF FOG, 5G AND MPTCP: THE REFERENCE 5G FOGRAN TECHNOLOGICAL PLATFORM

The reason why the convergence of the (aforementioned) three pillar paradigms of Fog Computing, 5G Communication and MPTCP is expected to enable fast seamless migrations of VMs stems out from their native features, that are, indeed, complementary (see Table 1 for a synoptic win-to-win overview).

Fog Computing (FC) is formally defined as a model for enabling the *pervasive* local access to a centralized pool of configurable computing/networking resources that can be rapidly provisioned and released in an elastic (e.g., on-demand) basis. Proximate resource-limited mobile devices access these facilities over a *single-hop* wireless access network, in order to reduce the communication delay and, then, support delay and delay-jitter sensitive VM migrations. As pointed in the first column of Table 1, the main native features retained by the Fog paradigm are [1]: (i) edge location and context awareness; (ii) pervasive spatial deployment; and, (iii) support for mobile energy-saving live migration of VMs.

Thank its sub-millisecond network latencies, the *5G Communication* paradigm provides the "right" means to support the wireless access to the virtualized resources hosted by Fog nodes. In fact, the 5G paradigm retains, by design, the following main native features (see the second column of Table 1) [4]: (i) ultra-low access latencies; (ii) on-demand provisioning of the wireless bandwidth; (iii) multiplexing, isolation and elastic scaling of the available physical communication resources; and, (iv) support for heterogeneous Radio Access Technologies (RATs) and multi-NIC mobile devices.

Thank to its support of multiple RATs, the 5G paradigm is the ideal partner of the *MPTCP*, whose native goal is to provide multi-homing (e.g., radio bundling) to mobile devices equipped with multiple heterogeneous WNICs. This means that, by design, MPTCP supports simultaneous data paths over *multiple* radio interfaces under the control of a *single* Transport-layer connection. Doing so, MPTCP attains (see the third column of Table 1) [2]: (i) bandwidth aggregation; (ii) robustness against mobility and/or fading-induced connection failures; (iii) backward compatibility with the legacy Single-Path TCP (SPTCP); and, (iv) load balancing of the migrated traffic.

The convergence of these three paradigms leads to the 5G FOGRAN technological platform of Fig. 1, that constitutes the reference scenario of this paper.

According to the emerging FOGRAN paradigm [6]–[8], in this scenario, a mobile device that is equipped with the MPTCP protocol stack establishes multiple connections to the serving Fog node by turning-ON and using in parallel its native WNICs (see the device in the top part of Fig. 1). Since the MPTCP is backward compatible with the SPTCP, a mobile device equipped with a single WNIC may continue to use legacy SPTCP connections (see the device in the bottom part of Fig. 1). The serving Fog node employs hypervisor-based virtualization technology, in order to deliver computational resources in the form of a VM that runs atop a cluster of networked physical servers hosted by the Fog node [3]. The hosted VM and the hosting Fog node communicate through a Virtual NIC (VNIC), that typically emulates in software an Ethernet switch [5]. According to the FOGRAN paradigm, the receive antennas of Fig. 1 (also referred to as Remote Radio Heads (RRHs)) perform only the Radio Frequency (RF) processing of the received data streams (e.g., power amplification, A/D and D/A conversions and sampling), while: (i) mo/demodulation and co/decoding base-band operations; (ii) MAC multiplexing/de-multiplexing and forwarding of the data flows; (iii) Network-layer routing; and, (iv) elastic resource allocation of the available communication bandwidth and resources, are collectively performed by the 5G Network Processor of Fig. 1. Specifically, according to the 5G paradigm (see Table 1), the 5G Network Processor multiplexes among the migrating VMs the available network physical resources by: (i) performing Network Function Virtualization (NFV); and, then, (ii) instantiating Virtual Base Stations (VBSs) atop the underlying 5G physical network (see Section VIII of [8]). The connection between the distributed RRHs and the centralized 5G Network Processor is provided by the front-haul section of Fig. 1, that is typically built up by ultra-broadband high-capacity optical-fiber channels [8].

In order to be robust against the connection failures (possibly) induced by wireless fading and/or device mobility, *pre-copy* migration [9] is the technique adopted by the wireless devices of Fig. 1 for performing VM offloading over the underlying 5G FOGRAN.

B. TACKLED PROBLEM, MAIN CONTRIBUTIONS AND ORGANIZATION OF THE PAPER

By referring to the 5G FOGRAN scenario of Fig. 1, the topic of this contribution is the settable-complexity energy-efficient dynamic management of the MPTCP bandwidth

TABLE 1: Native features of the pillar Fog, 5G and MPTCP paradigms and their synergic interplay.

Fog		5G		MPTCP	
Edge location and context awareness	Lasting one-hop from the served devices, Fog nodes may exploit context-awareness for the support of delay-sensitive VM migration	Ultra-low access latency	Sub-millisecond access latencies are obtained by combining massive MIMO, millimeter-wave, micro-cell and interference-suppression technologies	Bandwidth aggregation	By utilizing in parallel the available bandwidth of each path, MPTCP connections may provide larger throughput than SPTCP connections
Pervasive deployment	Fog nodes are a capable to support pervasive services, such as community services	On demand resource provisioning	Wireless bandwidth is dynamically provided an on-demand and per-device basis	Robustness	Path failure is recovered by shifting the traffic from the failed path to the remaining active ones
Virtualized environment	Software clones of the served wireless devices are dynamically bootstrapped onto the Fog servers as VMs	Resource multiplexing and isolation	Processing of the received flows is performed by a centralized Network Processor, that performs resource multiplexing/isolation on a per-flow basis through NFV	Backward compatibility	MPTCP connections utilize the same sockets of the standard SPTCP ones. Hence, SPTCP-based applications may continue to run unchanged under MPTCP
Mobility support	Fog nodes may exploit single-hop short-range *IEEE802.11x* links for enabling VM migrations from the mobile devices	Multi-RAT support	Multiple short/long-range WiFi/cellular networking technologies are simultaneously sustained, in order to support multi-NIC wireless devices	Load balancing	By a balanced splitting of the overall traffic over multiple paths, energy saving may be attained
Support for live VM migration	Wireless devices may seamlessly offload to serving Fog nodes their running applications in the form of VMs				
Energy efficiency	Wireless devices may save energy by exploiting nearby Fog nodes as surrogate servers				

for the Quality of Service (QoS) pre-copy live migration of VMs. The target is the minimization of the network energy wasted by each wireless device for the migration of own VMs under six hard constraints, which are typically induced by the desired implementation complexity-vs.-performance tradeoff.

Specifically, the first constraint regards the settable implementation complexity of the proposed bandwidth manager. In order to (shortly) introduce it, we point out that the pre-copy migration technique requires, by design, that the memory footprint of the VM is migrated over a number I_{MAX} of time rounds [9], whose transmission rates may be optimized in a more or less fine way, in order to save networking energy (see Fig. 4 in the sequel). A *per-round* optimization of the migration rates leads to the *maximum* saving of the network energy but also entails the *maximum* implementation complexity of the resulting bandwidth manager. Hence, the first constraint concerns the fact that, in order to limit the resulting implementation complexity, *only* a subset Q out of the available I_{MAX} migration rates is allowed to be dynamically updated.

The second and third constraints limit in a hard way the total migration time and downtime (e.g., the service interruption time) of the migrating VM. These constraints allow to support delay and delay-jitter sensitive applications run by the migrating VM. The forth constraint limits the maximum bandwidth available for the VM migration and it is enforced by the bandwidth allocation policy implemented by the 5G Network Processor of Fig. 1. The fifth constraint upper bounds the maximum slowdown (e.g., the maximum stretching of the execution time) that is tolerated by the migrated application. Finally, the last constraint enforces the convergence of the iteration-based migration process by introducing a hard bound on the feasible speed-up factor, e.g., the minimum ratio between the volumes of data migrated over two consecutive rounds (see also Fig. 4 in the sequel).

As matter of these facts, we anticipate that the major contributions of this paper are the following ones:

1) Being the MPTCP a novel paradigm that is not still fully standardized, various Congestion Control (CC) algorithms are currently proposed in the literature, in order to meet various application-depending tradeoffs among the contrasting targets of fairness, quick responsiveness and stable behavior [10]. Hence, since

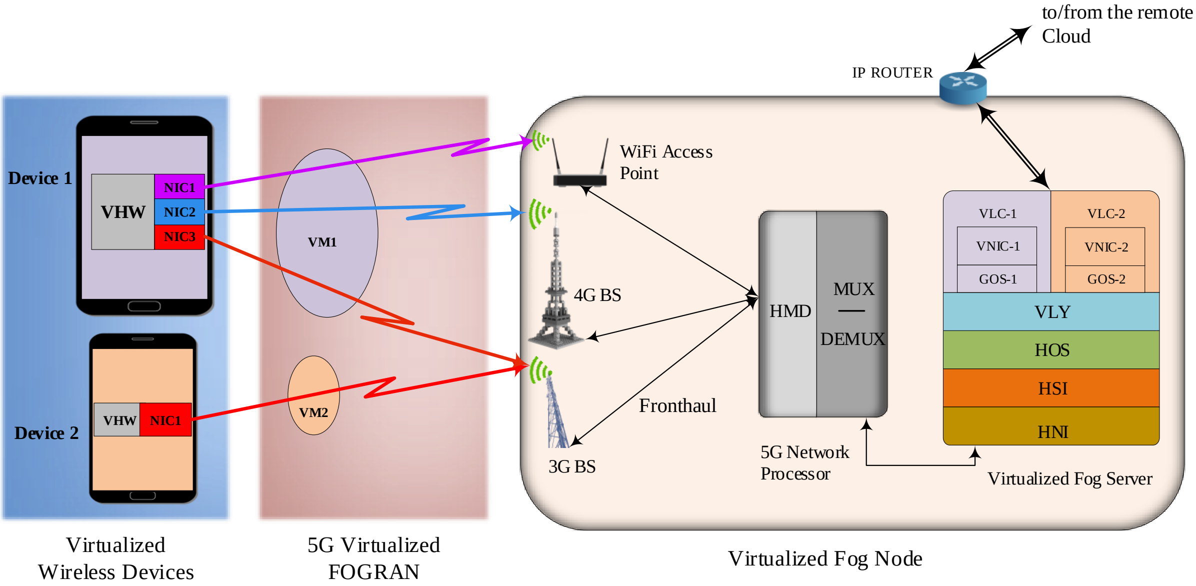


FIGURE 1: The considered virtualized scenario over a 5G access network. VHW = Virtualized HardWare; NIC = Network Interface Card; VM = Virtual Machine; BS = Base Station; HMD = Heterogeneous Mo-Demodulators; MUX = MUltipleXer; DEMUX = DEMUltipleXer; HNI = Hardware Network Infrastructure; HSI = Hardware Server Infrastructure; HOS = Host Operating System; RAN = Radio Access Network; VLY = Virtualization LaYer; VNIC = Virtual Network Interface Card; VCL = Virtual Clone; GOS = Guest Operating System; $\longleftrightarrow$ = Wired Connection.

the energy-vs.-transport rate profiles of the device-to-fog connections of Fig. 1 may depend also on the specifically adopted CC algorithm, we carry out a formal analysis of the power and energy consumptions of a number of MPTCP CC algorithms under the 5G FOGRAN scenario of Fig. 1. This is done for both cases of unbalanced and balanced MPTCP connections. The key result of this analysis is a *unified parametric* formula for the consumed dynamic power-vs.-transport rate profile that applies to *all* considered MPTCP CC algorithms, as well as to the newReno CC algorithm of the standard SPTCP.

2) On the basis of the obtained power profile of the wireless MPTCP connections, we develop a related delay-vs.-energy analysis, in order to formally characterize the operating conditions under which VM migration over the 5G FOGRAN of Fig. 1 allows the mobile device to really save energy. Interestingly, the carried analysis *jointly* accounts for the CPU and WNIC energies, as well as for the migration time, computing time and migration bandwidth.
3) By leveraging the obtained formulae for the energy-vs.-transport rate profile of the MPTCP wireless connections, we pass to formalize the problem of the minimum-energy live migration of VMs over the 5G FOGRAN of Fig. 1. For this purpose, we formulate the problem of the minimum energy consumption of the wireless device under limited migration time and downtime in the form of settable-complexity (non-convex) optimization problem, e.g., the minimum-energy Settable Complexity Bandwidth Manager (SCBM) optimization problem. Its formulation is *general enough* to embrace both MPTCP and SPTCP-based 5G FOGRAN scenarios, as well as in-band and out-band live migrations of VMs.
4) The state of the utilized MPTCP connection of Fig. 1 may be subject to fast (and, typically, unpredictable) fluctuations, that may be induced by: (i) channel fading; (ii) device mobility; and: (iii) time-varying behavior of the migrating application. Hence, motivated by this consideration, we develop an *adaptive* version of the resulting SCBM, which is capable to quickly react to the fluctuations of the state of the underlying MPTCP connection *without* requiring any form of (typically, unreliable and error-prone) forecasting.
5) We present the results of extensive numerical tests, in order to check and compare the actual energy-vs.-migration delay performances of the proposed adaptive MPTCP-based SCBM. Overall, we anticipate that the carried out tests lead to two first main insights. First, the implementation complexity of the SCBM *increases* in a linear way with the (aforementioned) number Q of updated transmission rates. At the same time, both the energy consumptions and times of convergence to the steady-state *decrease* for increasing values of Q and approach their global minima for values of Q quite low and typically limited up to $Q = 1 \div 3$. This supports the conclusion that the proposed adaptive SCBM is capable to attain a good performance-vs.-implementation complexity tradeoff. Second, in the carried out tests,

the energy savings of the proposed SCBM over the state-of-the-art one in [9] currently implemented by legacy Xen and KVM-based hypervisors [11], [12] are typically over 25%, and reaches 70% under strict limits on the allowed downtimes.

A last contribution concerns the MPTCP-vs.-SPTCP energy performances under the considered hard constraints on the migration times and downtimes. In principle, due to the simultaneous powering of multiple WNICs, the network power required to sustain a MPTCP connection is (obviously) larger than the one needed by the corresponding SPTCP connection. However, due to the bandwidth aggregation effect, the opposite conclusion hold for the resulting migration time. Hence, since the migration energy equates the power-by-migration time product, an interesting still open question [13], [14] concerns the actual energy reductions possibly attained by MPTCP paradigm over the SPTCP one under the migration scenario of Fig. 1. At this regard, we point out that:

6) a last set of carry out numerical tests stress that the energy savings of the MPTCP connections with respect to the corresponding SPTCP ones may be significant in supporting the proposed SCBM. Specifically, the tested savings may reach 30% and 70% with respect to SPTCP setups that utilize only WiFi and 4G connections, when the tested migration scenario meets at least one of the following operating conditions: (i) the size of the migration VM is quite large (e.g., we say, of the order of some tens of Megabits); (ii) the tolerated downtimes are low (e.g., less than 200 ms); (iii) the dirty rate (that is, the rate of memory-write operations) of the migrated application is not negligible (e.g., the dirty rate-to-available migration bandwidth ratio is larger than 10%).

The rest of this paper is organized as follows. After a review of the main related work of Section II, in Section III, we shortly summarize the basic features of the pre-copy live migration, in order to point out its inherent failure recovery capability. By leveraging on the (aforementioned) unified characterization of the power-vs.-rate profile of the currently considered MPTCP CC algorithms of Section IV, in Section V, we develop a delay-vs.-energy analysis, that aims to formally feature the operating conditions under which VM migration should be performed by the wireless device. Sections VI and VII are devoted to the formal presentation of the afforded constrained minimum-energy SCBM optimization problem, its feasibility conditions and the pursued solving approach, respectively. Afterward, in Section VIII, we develop the adaptive implementation of the proposed SCBM and discuss a number of related implementation aspects. Section IX is devoted to the presentation of the carried out numerical tests and performance comparisons under synthetic/real-world applications and static/dynamic MPTCP wireless connections. The conclusive Section X reviews the main presented results and points out some hints for future research. Auxiliary analytical results are provided in the final Appendix.

Regarding the main adopted notation, the superscript: $\vec{(\cdot)}$ denotes vector parameters, the symbol: $\triangleq$ indicates a definition, $\log$ is the natural logarithmic, while $\delta(y)$ is the Kronecker's delta (i.e., $\delta(y) = 1$ and $\delta(y) = 0$, at $y = 0$ and $y \neq 0$, respectively). Table 2 lists the main acronyms used in the paper.

TABLE 2: List of the main acronyms used in the paper.

Acronym	**Description**
CC	Congestion Control
CWND	Congestion Window
EWTCP	Ensemble Wireless TCP
FC	Fog Computing
MCC	Mobile Cloud Computing
MPTCP	MultiPath TCP
NFV	Network Function Virtualization
PeCM	Pre-copy Migration
PoCM	Post-copy Migration
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RRH	Remote Radio Head
RTT	Round Trip Time
SaCM	Stop-and-Copy Migration
SCBM	Settable Complexity Bandwidth Manager
SPTCP	SinglePath TCP
VM	Virtual Machine
WNIC	Wireless Network Interface Card

II. RELATED WORK

During the last years, the utilization of live VM migration as a network primitive functionality for attaining resource multiplexing, failure recovery and (possibly) energy reduction in centralized/distributed wired/wireless environments received an increasing attention both from industry and academy. An updated review on the main proposed solutions, open challenges and research directions is provided, for example, in [5]. However, at the best of the authors' knowledge, the wireless bandwidth management problem afforded by this paper under the 5G FOGRAN scenario of Fig. 1 deals with a still quasi unexplored topic, as also confirmed by an examination of [5] and references therein. In fact, roughly speaking, the published work mostly related to the afforded topic embraces four main research lines, namely: (i) the management of the network resources for the live VM migration; (ii) the utilization of the MPTCP for the live VM migration; (iii) the analysis and test of the MPTCP throughput in wireless/mobile scenarios; and, (iv) the development of Middleware platforms for the support of live VM migration.

Regarding the first research line, the current solution for the management of the migration bandwidth implemented by state-of-the-art hypervisors (like, for example, Xen, VMware and KVM) is the *heuristic* one firstly proposed in [9]. According to this heuristic solution, the migration bandwidth is linearly increased over consecutive migration rounds up to its maximum allowed value. Since the (single) goal of this heuristic is to reduce the final downtime, it neglects at all any energy-related performance index and does not enforces any constraints on the total migration time and/or downtime. Both these aspects are, indeed, accounted for by the authors of the (recent) contribution in [15]. However, this last contribution focuses only on *intra-data center* VM migrations over cabled Ethernet-type connections under legacy single-path newReno TCP. As a consequence, the solution in [15]: (i) does *not* allow to tradeoff the implementation complexity versus the resulting energy performance; and, (ii) it implements, by design, an adaptive mechanism that mainly aims to attain stable behavior in the steady-state *instead of* quick reaction to the abrupt mobility-induced changes of the state of the underlying wireless connection. As consequence of these different design criteria pursued in [15], we anticipate that, in the considered wireless application scenario of Fig. 1, the energy reductions attained by the here proposed adaptive SCBM over the solution in [15] may be significant (e.g., up to 50% under some harsh wireless scenarios).

Recent contributions along the (previously mentioned) second research line are reported in [16]–[18]. Specifically, the authors of [16] propose a nearby VM-based cloudlet for boosting the performance of real-time resource-stressing applications. In order to reduce the delay induced by the service initiation time, the authors of [16] develop an inter-cloudlet MPTCP-supported migration mechanism, that exploits the predicted mobility patterns of the served mobile devices in a pro-active way. Interestingly, almost zero downtimes are experienced at the destination cloudlets after the VM migrations are completed. The contribution in [17] proposes an adaptive queue-management scheduler, in order to exploit at the maximum the bandwidth aggregation capability offered by MPTCP when intra-data center live migration of VM must be carried out under delay-constraints. The topic of [18] is the exploitation of the aggregated bandwidth offered by MPTCP connections for the inter-fog wireless migration of Linux containers. For this purpose, a suitable version of the so-called Checkpoint/Restore In User-space (CRIU) migration technique is developed and its delay-performances are tested through a number of field trials. Overall, like our paper, the focus of all these contributions is on the effective exploitation of the MPTCP bandwidth aggregation capability, in order to reduce the VM migration times. However, unlike our paper, these contributions: (i) do *not* afford the energy-related aspects; and, (ii) do *not* consider the optimal management of the migration bandwidth under *hard* QoS-induced delay-constraints.

Testing the MPTCP throughput under wireless/mobile scenarios is the main topic of the contributions in [13], [14], [19]–[21]. Specifically, the authors of [13] investigate the interplay between traffic partitioning and bandwidth aggregation under various MPTCP-supported wireless scenarios and, then, propose a per-component energy model, in order to quantify the energy consumed by the device's CPU during the migration process. The focus of [19] is on the 3G/WiFi handover under MPTCP and related energy measurements, while [14] and [20] report the results of a number of tests on the delay-vs.-energy MPTCP performance under different flow sizes and mobility scenarios. Lastly, [21] proposes a power model for MPTCP and tests its energy consumption under different file sizes to be transferred. Overall, like our contribution, the common topic of all these papers is on the measurements of the power/energy performance of MPTCP connections under wireless/mobile outdoor scenarios. However, unlike our work, the focus of these papers is *not* on the live migration of VMs and/or on the effect of the various MPTCP CC algorithms on the energy performance of the underlying MPTCP connections.

Finally, [3] reports a quite comprehensive overview on a number of recent contributions, whose common topic is the development of software middleware platforms for the real-time support of traffic offloading. Among these contributions, we (shortly) review those reported in [22]–[29], that mainly match the considered virtualized scenario of Fig. 1. A synoptic examination of these contributions allows us to enucleate the basic building blocks and related data paths that are shared by all them, as sketched by the reference virtualized architecture of Fig. 2.

Hence, by referring to Fig. 2, the MAUI [22] and CloneCloud [23] frameworks develop and prototype in software two middleware platforms, that allow mobile devices to offload their virtualized tasks directly to public cloud data centers through cellular/WiFi connections. The common feature of the middleware virtualized platforms developed in [24]–[27] is to be *cloudlet*-oriented. This means that the middleware layers of all these platforms rely on single-hop WiFi-based links, in order to perform fine/coarse-grained traffic offloading to nearby small-size data centers, generally referred to as cloudlets. The Mobile Network Operator Cloud (MNOC) is, indeed, the common focus of the contributions in [28], [29]. These contributions consider a framework in which mobile devices exploit cellular 3G/4G connections, in order to offload their virtualized tasks to data centers that are directly managed by mobile network operators. Overall, like our work, all these contributions consider virtualized middleware-layer platforms for the resource augmentation of resource-limited wireless devices. However, unlike our work, the main focus of these contributions is on the design and implementation of middleware-layer software for the support of traffic migration, and they do not consider the problem of the delay-constrained and energy-efficient optimized management of the migration bandwidth.

III. BASIC LIVE MIGRATION TECHNIQUES – A SHORT OVERVIEW

Live VM migration allows a running VM to be transferred between different physical machines without halting the migrated application. In principle, there are four main techniques for live VM migration, namely, Stop-and-Copy Migration (SaCM), Pre-Copy Migration (PeCM), and Post-Copy Migration (PoCM). They trade-off the volume of migrated data against the resulting downtime. These techniques rely on the implementation of at least one of the following three phases [5]:

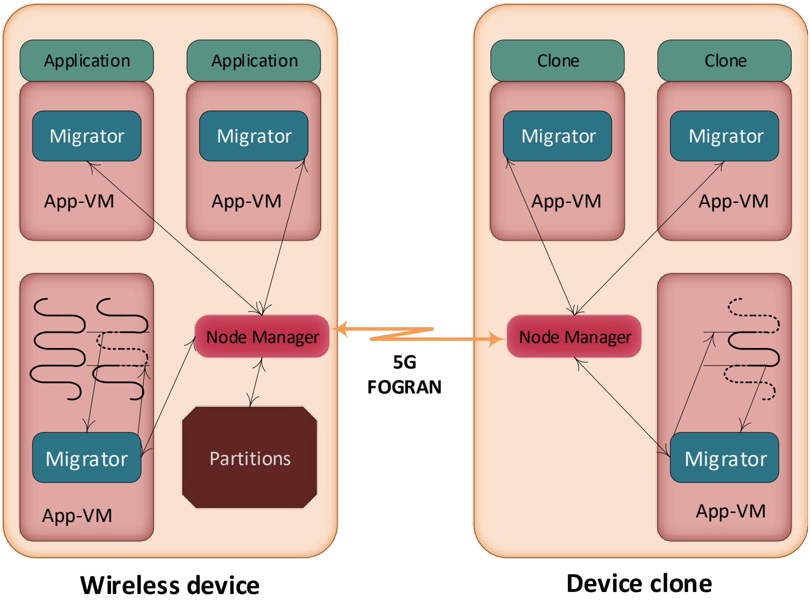


FIGURE 2: The reference virtualized architecture considered for the middleware support of traffic offloading.

(i) **Push phase**: the source device transfers to the destination device the memory image (e.g., the RAM content) of the migrating VM over consecutive rounds. To ensure consistency, the memory pages modified (e.g., dirtied) during this phase are re-sent over multiple rounds;

(ii) **Stop-and-copy phase**: the VM running at the source device is halted and the lastly modified memory pages are transferred to the destination;

(iii) **Pull phase**: the migrated VM begins to run on the destination device. From time to time, the access to the memory pages still stored by the source device is accomplished by issuing page-fault interrupts.

By design, the SaCM technique utilizes only the stop-and-copy phase. This guarantees that the volume of the migrated data equates the memory size of the migrated VM, but it generally induces long downtimes [5]. The PeCM technique implements both the push and stop-and-copy phases of the migration process. It guarantees finite migration times, tolerable downtimes and robustness against the (possible) failures of the communication link. However, it induces overhead in the total volume of the migrated data, which may be substantial under write-intensive applications [5]. The PoCM technique is composed by the stop-and-copy and pull phases of the migration process. Since only the I/O and CPU states of the source device are transferred to the destination device during the initial stop-and-copy phase, the experienced downtime is limited and no data overhead is induced. However, the resulting total migration time is, in principle, undefined and, due to the page-fault interrupts issued during the pull phase, the slowdown experienced by the migrated application may be substantial. We anticipate that the optimal bandwidth manager developed in this paper may be applied under all the mentioned migration techniques. However, in order to speed up its presentation, in the sequel, we focus on the PeCM as reference technique. The main reasons behind this choice are that: (i) PeCM is the migration technique currently implemented by a number of commercial hypervisors, such as, Xen, VMware and KVM; and, (ii) the bandwidth management framework provided by the PeCM technique is general enough to embrace those featured by the SaCM and PoCM techniques.

From a formal point of view, the PeCM technique is the cascade of the six stages reported in Fig. 3, namely:

1) **Pre-migration**: the VM to be migrated is built up on the source device and the destination machine is selected on the destination server. This stage spans T_{PM} seconds.
2) **Reservation**: the computing/communication/storage/memory physical resources are reserved at the destination server by instantiating a large enough VM container. T_{RE} (s) is the duration (in seconds) of this stage.
3) **Iterative pre-copy**: this stage is composed by $(I_{MAX}+1)$ rounds and spans T_{IP} seconds. During the initial round (e.g., at round #0), the full memory content of the migrating VM is sent to the destination server. During the subsequent I_{MAX} rounds (e.g., from round #1 to round #I_{MAX}), the memory pages modified (e.g., dirtied) during the previous round are re-transferred to the destination server (see Fig. 4).
4) **Stop-and-copy**: the migrating VM is halted and a final memory-copy round (e.g., round #$(I_{MAX}+1)$) is performed (see Fig. 4). This last round spans T_{SC} seconds.
5) **Commitment**: the destination server notifies that it has received a right copy of the migrated VM. T_{CM} (s) is

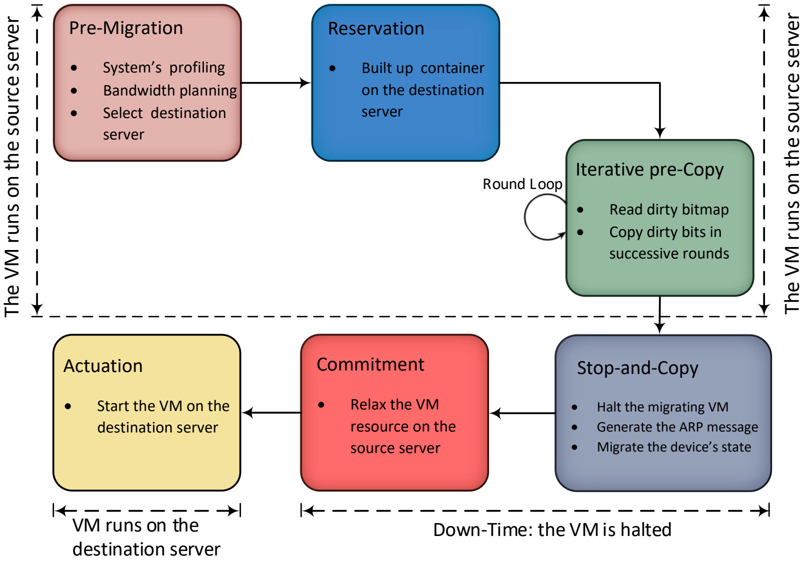

Pre-Migration
System's profiling
Bandwidth planning
Select destination server
Reservation
Built up container on the destination server
Iterative pre-Copy
Read dirty bitmap
Copy dirty bits in successive rounds
Round Loop
The VM runs on the source server
The VM runs on the source server
Actuation
Start the VM on the destination server
Commitment
Relax the VM resource on the source server
Stop-and-Copy
Halt the migrating VM
Generate the ARP message
Migrate the device's state
VM runs on the destination server
Down-Time: the VM is halted

FIGURE 3: The six stages of the Pre-Copy Migration (PeCM) technique.

the duration of this stage.

6) **Re-activation**: the I/O resources and IP address are re-attached to the migrated VM on the destination server. T_{AT} (s) is the needed time.

A. FORMAL DEFINITION OF THE MIGRATION DELAYS

From a formal point of view, the total migration time T_{tot} (s) is the overall duration:

$$T_{tot} \triangleq T_{PM} + T_{RE} + T_{IP} + T_{SC} + T_{CM} + T_{AT}, \qquad (1)$$

of the (aforementioned) six stages of Fig. 3, while the down-time:

$$T_{DT} \triangleq T_{SC} + T_{CM} + T_{AT}, \qquad (2)$$

is the time required for the execution of the corresponding last three stages. From an application point of view, T_{tot} in Eq. (1) is the time over which the source and destination servers must be synchronized, while T_{DT} in Eq. (2) is the period over which the migrating VM is halted and the user experiences a service outage. Let R (Mb/s) be the transmission rate (measured at the Transport layer) during the third and fourth stages of the migration process, that is, the migration bandwidth. Since, by definition, only T_{IP} and T_{SC} depend on R, while all the remaining migration times in Eqs. (1) and (2) play the role of constant parameters, in the sequel, we focus on the evaluation of the (already defined) stop-and-copy time T_{SC} and the resulting memory migration time T_{MT}, formally defined as:

$$T_{MT} \equiv T_{MT}(R) \triangleq T_{IP}(R) + T_{SC}(R). \qquad (3)$$

Hence, T_{MT} is the time needed for completing the memory transferring of the migrating VM, e.g., the duration of the performed $(I_{MAX} + 2)$ memory-copy rounds of Fig. 4.

Since the PeCM technique performs the iterative pre-copy of dirtied memory pages over consecutive rounds (see Fig. 4), let V_i (Mb) and T_i (s), $i = 0, \ldots, I_{MAX}+1$, be the volume of the migrated data and the time duration of the i-th migration round of Fig. 4, respectively. By definition, V_0 and T_0 are the memory size M_0 (Mb) of the migrating VM and the time needed for migrating it during the 0-th round, respectively (see the leftmost part of Fig. 4). Hence, after indicating by: $\overline{w}$ (Mb/s) the (average) memory dirty rate of the migrating application (e.g., the per-second average number of memory bits which are modified by the migrating application), from the reported definitions we have that:

$$V_i \triangleq \overline{w}\, T_{i-1}, \quad i = 1, \ldots, I_{MAX} + 1, \qquad (4)$$

with $V_0 \equiv M_0$, and also:

$$T_i \triangleq \frac{V_i}{R_i} \equiv \frac{\overline{w}}{R_i} T_{i-1} = M_0\, \overline{w}^i \left(\prod_{m=0}^{i} R_m^{-1} \right), \qquad (5)$$

for $i = 0, \ldots, I_{MAX} + 1$, with $T_0 \equiv \frac{M_0}{R_0}$. As a consequence, we have that (see Eq. (3)):

$$T_{MT}(R) \equiv \sum_{i=0}^{I_{MAX}+1} T_i = M_0 \left[\sum_{i=0}^{I_{MAX}+1} \overline{w}^i \left(\prod_{l=0}^{i} R_l^{-1} \right) \right], \qquad (6)$$

and then (see Eq. (5)):

$$T_{SC}(R) \equiv T_{I_{MAX}+1} = M_0\, \overline{w}^{I_{MAX}+1} \left(\prod_{m=0}^{I_{MAX}+1} R_m^{-1} \right). \qquad (7)$$

In order to speed up the paper readability, the main used symbols and their meanings are listed in Table 3.

IV. POWER AND ENERGY ANALYSIS OF MPTCP WIRELESS CONNECTIONS

With reference to the 5G FOGRAN environment of Fig. 1, main goal of this section is to develop a formal analysis of

TABLE 3: Main symbols used in the paper and their meanings.

Symbol	Meaning
N	Number of WNICs equipping the wireless device
I_{MAX}	Maximum number of migration pre–copy rounds
i	Round index, with $i = 0, \dots, (I_{MAX} + 1)$
j	Integer-valued path-index of the MPTCP connection, with $j = 1, \dots, N$
n	Integer-valued iteration index
$\overline{w}$ (Mb/s)	Average memory dirty-rate of the migrated application
R (Mb/s)	Migration bandwidth utilized by the MPTCP
$\mathcal{P}$ (W)	Dynamic network power consumed by the MPTCP
M_0 (Mb)	Memory size of the migrated VM
Δ_{MT} (s)	Maximum tolerated migration time
Δ_{DT} (s)	Maximum tolerated downtime
$\widehat{R}$ (Mb/s)	Maximum migration bandwidth of the MPTCP
$\mathcal{E}_{setup}$ (J)	Total energy consumed by the mobile device for the parallel setup of its WNICs
$\mathcal{E}_{tot}$ (J)	Total setup-plus-dynamic network energy consumed by the mobile device for the VM migration

the dynamic power-vs.-transport rate profile of the MPTCP connections that accounts (in a unified way) for the effects of the various CC algorithms currently considered in the literature [10]. We anticipate that the final formulae of the carried out analysis will be directly employed in Section VI, in order to formally define the objective function of the considered SCBM problem. In order to put the performed analysis under the right reference framework and speed-up its development, in the next two sub-sections, we shortly review some basic architectural features of the MPTCP protocol stack and related CC algorithms.

A. A SHORT REVIEW OF SOME KEY FEATURES OF THE MPTCP PROTOCOL STACK

Fig. 5 sketches the main components of the MPTCP-compliant protocol stack implemented by a wireless device that is equipped with $N \geq 2$ heterogeneous WNICs [2]. Each WNIC posses an own IP address at the Network layer[1], with $IP(j)$, $j = 1, \dots, N$, denoting the IP address of the j-th WNIC. The MPTCP layer is a backward-compatible modification of the standard SPTCP that allows a single transport layer data flow (e.g., a single transport layer connection) to be split across the N paths (also referred to as subflows) done available by the underlying Network layer. For this purpose, the Application and MPTCP layers communicate through a (single) socket that is labeled by a *single* port number [2]. After performing the connection setup through the exchange of SYN/ACK segments that carry out the MP_CAPABLE option, the running application opens the socket, which, in turn, starts the Slow-Start phase of a first TCP subflow. If required by the application socket, more sub-flows can be dynamically added (resp., removed) later by issuing the MP_JOIN/ADD_ADDR commands of the MPTCP protocol. Each opened sub-flow is pinned to an own WNIC and labeled

[1] Both the IPv4 and IPv6 Internet protocols may run under the MPTCP layer of Fig. 5 [2].

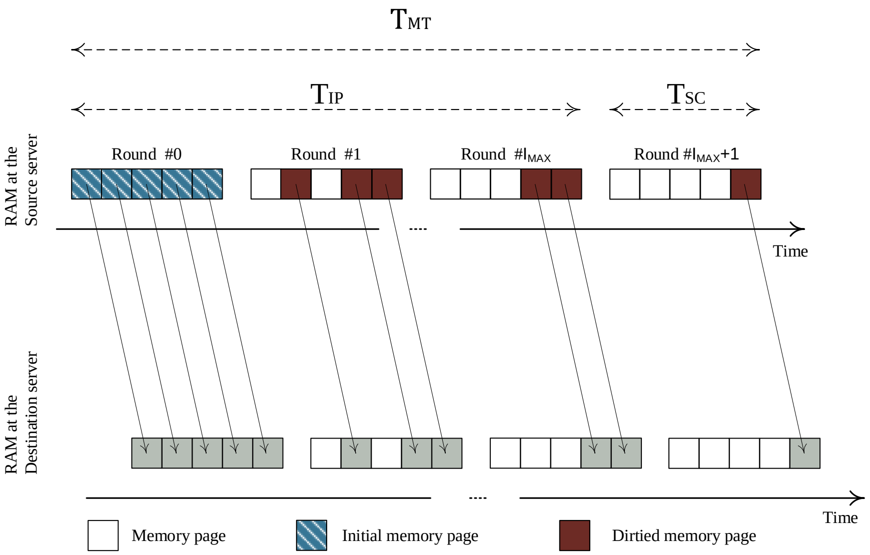


FIGURE 4: An illustrative time-chart of the iterative pre-copy scheme of the PeCM technique.

by the corresponding IP address (see Fig. 5).

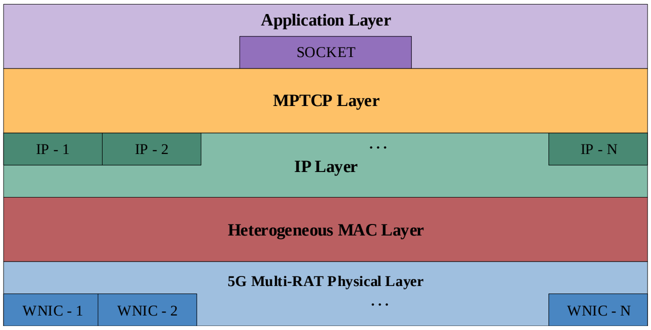


FIGURE 5: A sketch of the MPTCP protocol stack.

The MPTCP protocol uses two levels of sequence number for guaranteeing the in-order delivering of the transported, namely, a single connection-level sequence number and a set of subflow-level sequence numbers. The single connection-level sequence number is the data sequence number utilized by the running application. When the MPTCP sender starts to transmit in parallel over multiple subflows, each connection-level sequence number is suitably mapped onto a corresponding subflow sequence number. Doing in so, each subflow can send own data and process the corresponding ACK messages as a regular standing-alone SPTCP connection. The MPTCP receiver uses the connection-level sequence number for orderly reassembling the received subflows. An arrived data segment is marked as "in-sequence" if both its subflow and connection sequence numbers are in-order.

Out-of-sequence segments are temporarily stored by a connection-level buffer that equips the receive side of the MPTCP connection. This buffering introduces, in turn, harmful queue-induced delivering delays, that, in some limit cases, may fully offset the corresponding reduction in the transport delays gained by using MPTCP. Hence, out-of-sequence phenomena should be avoided as much as possible.

B. A SYNOPTIC COMPARISON OF THE MPTCP CONGESTION CONTROL ALGORITHMS

After exiting the setup phase, the state of the MPTCP connection enters the Slow Start (SS) phase. During this phase, each active subflow works independently from the others and applies the same SS algorithm as a regular SPTCP flow [2].

At the end of the SS phase, the MPTCP connection enters the Congestion Avoidance (CA) state, e.g., the steady-state. During this state, the behavior of the MPTCP connection is managed by the the corresponding CC algorithm, whose specific features make the MPTCP very different from the regular SPTCP. To begin with, the MPTCP sender implements and updates in parallel multiple mutually interacting Congestion WiNDows (CWNDs) for controlling the local traffic over each path, whilst the MPTCP receiver uses a single receive window [2]. The (possibly, coupled) updating of the CWNDs at the sender side is dictated by the actually adopted CC algorithm. A family of different MPTCP CC algorithms have been proposed over the last years, each one featured by a specific tradeoff among the contrasting targets of fair behavior, quick responsiveness in the transient-state and stable behavior in the steady-state (see [10] for a formal presentation of this specific topic).

However, according to the analysis presented in [10], the basic control mechanism implemented by the currently proposed CC algorithms may be shortly described as follows. Let $W_d(j)$, $j = 1, \ldots, N$, be the size (measured in Mb) of the CWND of the j-th subflow and let:

$$W_d \triangleq \sum_{j=1}^{N} W_d(j), \tag{8}$$

be the resulting size of the total CWND. On each subflow j, the MPTCP source increases (resp., decreases) the corresponding j-th CWND by a number of Megabits equal to: $I_c(j)$ (resp., $D_c(j)$) at the return of each ACK message (resp., at the detection of each segment loss). These increments and decrements are dictated by the following general expressions [10]:

$$W_d(j) = W_d(j) + I_c(j), \tag{9}$$

for each ACK on the j-th subflow, and

$$W_d(j) = W_d(j) - D_c(j), \tag{10}$$

for each segment loss on the j-th subflow.

After denoting by $RTT(j)$ the round-trip-time (measured in seconds) of the j-th subflow, let: $R(j) \triangleq W_d(j)/RTT(j)$ (Mb/s) be the corresponding net transmission rate (e.g., the throughput), so that:

$$R \triangleq \sum_{j=1}^{N} R(j), \tag{11}$$

is the resulting total transmission rate (e.g., the total throughput) of the overall MPTCP connection. Hence, as detailed in Table 4 [10], the specific feature of each CC algorithm depends on the way in which the increment/decrement terms in (9) and (10) are actually computed. For comparison purpose, the last row of Table 4 reports the basic CC features of the commodity NewReno SPTCP.

Detailed analysis and comparisons of the fairness/responsiveness/stability properties of the MPTCP CC algorithms of Table 4 are reported in [10] and, then, they will be not replicated here. We shortly limit to remark that: (i) all the considered CC algorithms react against per-subflow segment loss events by halving the current size of the corresponding per-subflow congestion window; (ii) the EWTCP algorithm applies the NewReno SPTCP increment policy on each subflow independently. For this reason, as also pointed out in [10] and [32], it is the most reactive in coping with the abrupt changes typically affecting the state of wireless/mobile MPTCP connections; (iii) the Semicoupled, Max and Balia algorithms introduce various forms of coupling in updating the sizes of the congestion windows, in order to be more fair (like as the Balia CC algorithm) and/or more stable in the

TABLE 4: Expressions of the increments/decrements of the per-flow congestion windows of some MPTCP CC algorithms [10]. SPTCP mantains, by design, a single subflow.

Acronym of the CC algorithm	Expression of $\mathbf{I_c(j)}$	Expression of $\mathbf{D_c(j)}$	Auxiliary notations
EWTCP [30]	$a/W_d(j)$	$W_d(j)/2$	$a = 1$
Semicoupled MPTCP [31]	$a/W_d, \forall j = 1, \ldots, N$	$W_d(j)/2$	$a = N$
Max MPTCP [32]	$\min\left\{\frac{1}{W_d(j)};\ \frac{\max_k\{W_d(k)/RTT^2(k)\}}{\left(\sum_{k=1}^{N} W_d(k)/RTT(k)\right)^2}\right\}$	$W_d(j)/2$	$a = N^2$
Balia MPTCP [10]	$\left(\frac{R(j)}{RTT(j)\left(\sum_{k=1}^{N} R(k)\right)^2}\right)\left(\frac{1+\gamma(j)}{2}\right)\left(\frac{4+\gamma(j)}{5}\right)$	$(W_d(j)/2) \times \min\{\gamma(j);\ 1.5\}$	$a = N^2;\ \gamma(j) = \frac{\max_k(R(k))}{R(j)}$
NewReno SPTCP	$1/W_d$	$W_d(j)/2$	$N = 1$

steady-state (like as the Semicoupled CC algorithm). Hence, these last CC algorithms appear to be more suitable for managing wired (possibly, multi-hop) MPTCP connections (see [2] and Section V of [10]).

C. UNIFIED ANALYSIS OF THE POWER-VS.-RATE PERFORMANCE OF THE MPTCP UNDER 5G SCENARIOS

The total network energy $\mathcal{E}_{tot}$ (J) consumed by the multiple WNICs equipping the wireless device of Fig. 1 during each VM migration process is the summation of a static (e.g., setup) part: $\mathcal{E}_{setup}$ (J), and a dynamic part: $\mathcal{E}_{dyn}$ (J). By design, $\mathcal{E}_{setup}$ does not depend on the set of the per-subflow transmission rates and equates the summation:

$$\mathcal{E}_{setup} \triangleq \sum_{j=1}^{N} \mathcal{E}_{setup}(j), \tag{12}$$

of the setup energies needed by the involved WNICs to establish and maintain the MPTCP connection [14]. The dynamic portion $\mathcal{E}_{dyn}$ of the consumed energy depends on the per-subflow transmission rates in (11) through the corresponding total dynamic power $\mathcal{P}$ (W), that, in turn, is given by the following summation:

$$\mathcal{P} \triangleq \sum_{j=1}^{N} \mathcal{P}(j), \tag{13}$$

where $\mathcal{P}(j)$, $j = 1, \ldots, N$, is the dynamic power consumed by the j-th WNIC for sustaining its transport rate $R(j)$ (see Eq. (11)). Now, the key point to be remarked is that the (aforementioned) MPTCP CC algorithms may introduce coupling effects, so that each $\mathcal{P}(j)$ may be a function of the *overall* spectrum: $\{R(k), k = 1, \ldots, N\}$ of transport rates, with the analytical form of the function that may depend, in turn, on *both* the subflow index j and the actually considered CC algorithm. Interestingly, a suitable exploitation of the expressions of Table 4 allows us to arrive at the following *unified* formula for the profile of the total dynamic power $\mathcal{P}$ in (13) as a function of transport rates $\{R(k), k = 1, \ldots, N\}$ (see the Appendix A for the derivation):

$$\mathcal{P} = \mathcal{A}\left(\sum_{j=1}^{N} \tau(j)\,(R_j)^c\right). \tag{14}$$

In Eq. (14), the expressions of $\mathcal{A}$, c and $\{\tau(j), j = 1, \ldots, N\}$ depend on the considered CC algorithms. They are detailed in Table 5, where: (i) MSS (Mb) is the maximum size of a MPTCP segment; (ii) $\alpha > 1$ is a dimensionless power exponent; (iii) the constant a is given by the forth column of the above Table 4; and, (iv) $\Omega(j)$, $j = 1, \ldots, N$, represents the noise power-to-coding gain ratio of the transmission path sustained by the j-th WNIC. According to Eq. (63) of the Appendix A, it is measured in $(W^{1/\alpha})$ and it is formally defined as follows:

$$\Omega(j) \triangleq Pr_{loss}(j)\left(\mathcal{P}(j)^{1/\alpha}\right), \tag{15}$$

where $Pr_{loss}(j)$ is the (dimensionless) segment loss probability experienced by the j-th subflow.

Regarding the derived dynamic power-vs.-transport rate formula in (14), two main remarks are in order. First, it is composed by a linear superposition of N power terms, each one involving the transport rate of a MPTCP subflow. This power-like form is compliant, indeed, with the results reported by a number of measurement/test-based studies [13], [14], [20], [21], [33]–[35]. Second, the key features of our formula in (14) are that: (i) it holds in general, e.g., regardless from the specifically considered CC algorithm; and, (ii) its parametric form allows us to formally account in a direct way for the effect induced on the consumed power by the actually considered CC algorithm (see Table 5).

D. THE CASE OF LOAD-BALANCED MPTCP CONNECTIONS

A direct inspection of Eqs. (14) and (15) leads to the conclusion that, when the N subflows are load balanced, e.g.:

$$R(1) = R(2) = \ldots = R(N) \equiv \frac{R}{N}, \tag{16}$$

then, the power-rate formula in (14) reduces to the following *monomial* one:

$$\mathcal{P} = K_0\,(R)^{\alpha}. \tag{17}$$

TABLE 5: Expressions of the parameters involved by the unified power-vs.-rate formula in (14) for the MPTCP.

Acronym of the CC algorithm	**Expression of $\mathcal{A}$**	**Expression of $\tau(j)$**	**Expression of c**
EWTCP [30]	1	$\left((RTT(j)\times\Omega(j))\,/\,\left(MSS\sqrt{2a}\right)\right)^{\alpha}$	2α
Semicoupled MPTCP [31]	$\left(\frac{\sum_{k=1}^{N} RTT(k)R(k)}{2aMSS^{2}}\right)^{\alpha}$	$(RTT(j)\times\Omega(j))^{\alpha}$	α
Max MPTCP [32]	$\left(\frac{\left(\sum_{k=1}^{N} RTT(k)\right)^{2}}{2aMSS^{2}\times\max_{k}\left(\frac{R(k)}{RTT(k)}\right)}\right)^{\alpha}$	$(RTT(j)\times\Omega(j))^{\alpha}$	α
Balia MPTCP [10]	$\left(\frac{\left(\sum_{k=1}^{N} RTT(k)\right)^{2}}{0.4aMSS^{2}\times\max_{k}(R(k))}\right)^{\alpha}$	$(RTT(j)^{2}\times\Omega(j))^{\alpha}$	α
NewReno SPTCP	1	$((RTT/MSS)\times(\Omega/2))^{\alpha}$	α

In Eq. (17), K_0 is measured in $\left(\frac{W}{(Mb/s)^{\alpha}}\right)$ and its formal expressions are reported in Table 6 for the (above considered) CC algorithms.

The expression in (17) is central for the future developments of our paper, and merits, indeed, three main remarks.

First, from a formal point of view, an examination of the general expressions reported in the 3-rd column of Table 5 points out that the equal-balanced condition in (16) is met when the values assumed by the j-indexed products: $RTT(j)\times\sqrt{\Omega(j)}$ (resp., the products: $RTT(j)\times\Omega(j)$) do *not* depend on the subflow index j under the EWTCP and Balia MPTCP (resp., under the Semicoupled and Max MPTCP). The measurement-based studies reported, for example, in [14], [19], [21], [32]–[34] support the conclusion that, in practice, these formal conditions are quite well met under 3G/4G cellular and WiFi connections, mainly because the average segment loss probability (resp., the average round-trip-time) of WiFi connections is typically one order of magnitude larger (resp., lower) than the corresponding one of 3G/4G cellular connections.

Second, an inspection of the expressions of Table 6 leads to the conclusion that, at $\alpha>1$, the K_0 power-coefficient in (17) scales down for increasing number N of the available load-balanced subflows as:

$$K_0\propto\left(\frac{1}{N^{\alpha-1}}\right). \qquad (18)$$

In the sequel, we refer to the scaling down behavior in (18) as the "multipath gain" of the equal-balanced MPTCP. Intuitively, it arises from the fact that, at $\alpha>1$, the dependence of the dynamic power $\mathcal{P}$ in (14) on each subflow transport rate is of convex-type, so that equal-balanced transport rates reduce the resulting total power consumption.

Third, unbalanced transport rates increase the chance of out-of-sequence segment arrivals at the receive end of the MPTCP connection, that, in turn, give arise to harmful queue-induced delivering delays (see the last remark of the above Section IV-A).

Overall, motivated by the above remarks, in the sequel, we directly focus on the equal-balanced case, and, then, we exploit the expression in (17) for the formal characterization of the resulting total migration energy. At this regard, we anticipate that the topic regarding the on-line (e.g., real-time) profiling of the K_0 and α parameters present in (17) will be afforded in Section VIII-A, whilst some additional remarks on the case of unbalanced MPTCP connections will be reported in Section X.

V. TO MIGRATE OR NOT TO MIGRATE – A DELAY-VS.-ENERGY ANALYSIS

The three-fold task of the migration module at the mobile device of Fig. 2 is to plan: (i) "which" VM to migrate; (ii) "when" (e.g., under which operating conditions) performing the migration; and, (iii) "how" manage the planned migration. Although the focus of this paper is on the third question, in this section we shortly address the first two questions. At this regard, we observe that the final goal of the mobile-to-fog migration process would be to reduce as much as possible both the execution time of the migrating application and the corresponding energy consumed by the mobile device.

In the sequel, we develop a time-energy analysis of the mobile-to-fog migration costs under the considered 5G FOGRAN scenario of Fig. 1, in order to formally characterize the traded-off operating conditions under which the VM migration is worth.

In order to characterize and compare the local-vs.-remote execution times of the migrating VM, let $\gamma\triangleq(w_{VM}/M_0)$ be the ratio between the workload w_{VM} (bit) to be processed by the migrating VM and its (previously) defined size M_0 (bit). Hence, after indicating by s_{com}^{Mob} (bit/s) the processing (e.g., computing) speed at the mobile device, the resulting time T_{exe}^{Mob} (s) for the local execution of the VM at the mobile device equates:

$$T_{exe}^{Mob}\triangleq\frac{w_{VM}}{s_{com}^{Mob}}\equiv\frac{\gamma M_0}{s_{com}^{Mob}}. \qquad (19)$$

However, after indicating by s_{com}^{Fog} (bit/s) the processing speed of the device clone at the Fog node of Fig. 1, when the migration is performed, the resulting execution time T_{exe}^{Fog} (s) at the Fog node is the summation:

$$T_{exe}^{Tog}=T_{tot}+\frac{\gamma M_0}{s_{com}^{Fog}}, \qquad (20)$$

of the total migration time in Eq. (1) and the execution time: $w_{VM}/s_{com}^{Fog} \equiv \gamma M_0/s_{com}^{Fog}$, of the migrated VM at the Fog node. As a consequence, the VM migration would reduce the execution time when $T_{exe}^{Mob} \geq T_{exe}^{Fog}$, that is, when the corresponding migration time T_{MT} in Eq. (3) meets the following upper bound:

$$T_{MT} \leq \max\left\{0; \gamma M_0 \left(\frac{1}{s_{com}^{Mob}} - \frac{1}{s_{com}^{Fog}}\right) - \left(T_{PM} + T_{RE} + T_{CM} + T_{AT}\right)\right\}, \tag{21}$$

where the $\max\{\cdot\}$ operator in (21) accounts for the fact that, by definition, T_{MT} is non-negative.

Passing to consider the energy consumptions induced by the local and remote workload processing, let $\mathcal{P}_{com}^{Mob}$ (W) be the power consumed by the mobile device when it runs at the (aforementioned) processing speed s_{com}^{Mob}. Hence, the energy $\mathcal{E}_{com}^{Mob}$ (J) wasted by the mobile device for the local execution of the workload w_{VM} equates:

$$\mathcal{E}_{com}^{Mob} = \mathcal{P}_{com}^{Mob}\left(\frac{w_{VM}}{s_{com}^{Mob}}\right) \equiv \mathcal{P}_{com}^{Mob}\left(\frac{\gamma M_0}{s_{com}^{Mob}}\right). \tag{22}$$

The corresponding energy $\mathcal{E}_{com}^{Fog}$ (J) consumed by the mobile device when the VM is migrated and remotely processed at the Fog node is the summation of three contributions. The first one is the already considered energy $\mathcal{E}_{tot}$ consumed by the device-to-fog migration. The second one accounts for the idle energy: $\mathcal{E}_{idle}^{Mob}$ (J) consumed by the mobile device in the idle state during the remote execution of the migrated workload at the Fog node. It equates: $\mathcal{E}_{idle}^{Mob} = \mathcal{P}_{idle}^{Mob}\left(w_{VM}/s_{com}^{Fog}\right) \equiv \mathcal{P}_{idle}^{Mob}\left(\gamma M_0/s_{com}^{Fog}\right)$, where $\mathcal{P}_{idle}^{Mob}$ (W) is the power consumed by the mobile device in the idle state. The third component accounts for the energy: $\mathcal{E}_{R}^{Mob}$ (J) consumed by the mobile device, when it receives the processed workload sent back by the Fog node. It equates: $\mathcal{E}_{R}^{Mob} = \mathcal{P}_{R}^{Mob}\left(\mathcal{T} M_0/R_D\right)$, where: (i) $\mathcal{P}_{R}^{Mob}$ (W) is the network power consumed by all WNICs of the mobile device under the receive operating mode; (ii) the positive and dimensionless coefficient $\mathcal{T}$ is the relative size of the processed data, so that the product: $\mathcal{T} M_0$ (bit) is the size of the processed data sent back by the Fog node; and, (iii) R_D (bit/s) is the aggregated download bandwidth used by the Fog node for the fog-to-device data transfer over the 5G FOGRAN of Fig. 1. Overall, we have that:

$$\mathcal{E}_{com}^{Fog} = \mathcal{E}_{tot} + \mathcal{E}_{idle}^{Mob} + \mathcal{E}_{R}^{Mob} \equiv \mathcal{E}_{tot} + \mathcal{P}_{idle}^{Mob}\left(\frac{\gamma M_0}{s_{com}^{Fog}}\right) + \mathcal{P}_{R}^{Mob}\left(\frac{\mathcal{T} M_0}{R_D}\right). \tag{23}$$

Therefore, the VM migration is energy saving when: $\mathcal{E}_{com}^{Mob} \geq \mathcal{E}_{com}^{Fog}$, that is, when the migration energy $\mathcal{E}_{tot}$ meets the following inequality (see Eqs. (22) and (23)):

$$\mathcal{E}_{tot} \leq \max\left\{0;\ M_0\left[\left(\frac{\mathcal{P}_{com}^{Mob}}{s_{com}^{Mob}} - \frac{\mathcal{P}_{idle}^{Mob}}{s_{com}^{Fog}}\right)\gamma - \frac{\mathcal{T}\mathcal{P}_{R}^{Mob}}{R_D}\right]\right\}, \tag{24}$$

where the $\max\{\cdot\}$ operator in (24) accounts for the non-negativity of $\mathcal{E}_{tot}$. As a consequence, VM migration is time (resp., energy) efficient when the memory migration time T_{MT} (resp., the migration energy $\mathcal{E}_{tot}$) meets the upper bound in Eq. (21) (resp., in Eq. (24)).

An examination of the bounds in (21) and (24) allow us to address three questions about "which" VM to migrate, "when" performing the migration and "how" manage the (planned) migration. At this regard, three main remarks are in order.

First, we observe that the (previously defined) parameters M_0, γ and $\mathcal{T}$ depend only on the migrating VM and the corresponding migrated application. Hence, about the question concerning "which" VM to migrate, the lesson supported by the bounds in (21) and (24) is that the migrated VM should offer large workload w_{VM} (e.g., large values of the product: $M_0\gamma$), but quite limited values of the size of the processed data sent back from the Fog node (e.g, low values of the $\mathcal{T}$ coefficient in Eq. (24)).

Second, about the question concerning "when" performing the migration, the reported bounds point out that the 5G FOGRAN infrastructures which are more effective for supporting VM migration should be equipped with (very) speed

TABLE 6: Expressions of K_0 in Eq. (17) for the the case of load-balanced MPTCP.

Acronym of the CC algorithm	**Expression of $\mathbf{K_0}$** (W/(Mb/s)$^\alpha$)
EWTCP [30]	$\left(\frac{1}{\sqrt{2a}\times N\times MSS}\right)^\alpha \times \left(\sum_{k=1}^{N}\left(RTT(k)\times\sqrt{\Omega(k)}\right)^\alpha\right)$
Semicoupled MPTCP [31]	$\left(\frac{\sqrt{\sum_{k=1}^{N} RTT(k)}}{N\times MSS\times\sqrt{2a}}\right)^\alpha \times \left(\sum_{k=1}^{N}\left(RTT(k)\times\Omega(k)\right)^{\alpha/2}\right)$
Max MPTCP [32]	$\left(\frac{\min_k(RTT(k))}{2a\times MSS^2}\right)^{\alpha/2} \times \left(\sum_{k=1}^{N}\left(RTT(k)\times\Omega(k)\right)^{\alpha/2}\right)$
Balia MPTCP [10]	$\left(\frac{1}{0.4a\times MSS^2}\right)^{\alpha/2} \times \left(\sum_{k=1}^{N}\left(RTT(k)\times\sqrt{\Omega(k)}\right)^\alpha\right)$
NewReno SPTCP	$((RTT/MSS)\times(\Omega/2))^\alpha$

computing servers (i.e., $s_{com}^{Fog} >> s_{com}^{Mob}$) and broad Fog-to-mobile download bandwidths (i.e., large R_D, see Eq. (24)). Furthermore, the mobile devices that receive more benefit from the migration process are those equipped with power-consuming slow CPUs (e.g., low processing speeds s_{com}^{Mob} and high computing powers $\mathcal{P}_{com}^{Mob}$; see Eqs. (21) and (24)) and power-efficient WNICs (e.g., low idle and receive powers $\mathcal{P}_{idle}^{Mob}$ and $\mathcal{P}_{R}^{Mob}$, respectively; see Eq. (24)). Typical values of $\mathcal{P}_{com}^{Mob}$, $\mathcal{P}_{idle}^{Mob}$, $\mathcal{P}_{R}^{Mob}$, s_{com}^{Mob} and R_D for a spectrum of mobile devices and wireless access technologies are reported, for example, in [33]–[35] and [40].

Finally, about the question concerning "how" managing the planned VM migration, we observe that, in principle, under a given operating scenario (e.g., at fixed values of the bounds in Eqs. (21) and (24)), task of the migration module of Fig. 2 is to manage the device-to-Fog migration bandwidth of Fig. 1, in order to reduce as much as possible both the memory migration time and migration energy in Eqs. (21) and (24). However, an examination of the (previously reported) Eq. (6) points out that lower values of T_{MT} require higher migration bandwidths that, in turn, give arise to higher migration energies. This forbids to pursue the *simultaneous* minimization of T_{MT} and $\mathcal{E}_{tot}$. Hence, after observing that stretching as much as possible the battery life of the wireless devices remains a main target of the 5G paradigm, in the sequel, we approach the question concerning "how" managing the planned VM migration by formulating and solving a settable-complexity constrained optimization problem that *dynamically* minimizes at run-time the migration energy $\mathcal{E}_{tot}$, while simultaneously upper bounding (in a hard way) the resulting memory migration time in (6). In principle, the resulting SCBM provides the (formally optimal) response to the question about "how" manage the planned migration in an energy and time efficient way.

VI. THE AFFORDED SETTABLE-COMPLEXITY MINIMUM-ENERGY BANDWIDTH OPTIMIZATION PROBLEM

We pass, now, to formalize the afforded SCBM constrained optimization problem. The idea behind the proposed SCBM is quite simple. By referring to Fig. 4, in addition to R_0 and $R_{I_{MAX}+1}$, we update Q out of the I_{MAX} rates of the pre-copy rounds that are evenly spaced apart by $S \triangleq \frac{I_{MAX}}{Q}$ rounds over the round index set: $\{1, 2, \ldots, I_{MAX}\}$. For this purpose, we perform the partition of the round index set: $\{1, 2, \ldots, I_{MAX}\}$ into Q not overlapping contiguous clusters of size S, as shown in Fig. 6.

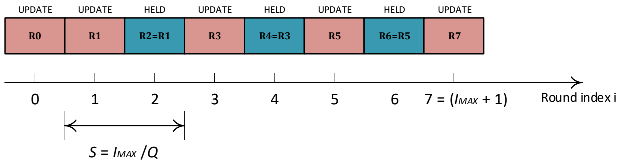


FIGURE 6: A sketch of the main idea behind the proposed SCBM.

The first rate R_{jS+1}, $j = 0, \ldots, Q-1$, of each cluster is updated, while the remaining $(S-1)$ rates are set to: R_{jS+1}, that is $R_i \equiv R_{jS+1}$, for $i = jS+2, jS+3, \ldots, (j+1)S$. Fig. 6 illustrates the framework of the updated/held migration rates for the case of $I_{MAX} = 6$ and $Q = 3$. In this example, R_0, R_1, R_3, R_5 and R_7 are the: $Q+2 = 5$ migration rates to be updated, while R_2, R_4 and R_6 are the: $(I_{MAX} - Q) = 3$ migration rates which are held, e.g., $R_2 \equiv R_1$, $R_4 \equiv R_3$ and $R_6 \equiv R_5$.

In order to formally introduce the SCBM, let I_{MAX} be the maximum number of the performed pre-copy rounds, so that the overall set of the $(I_{MAX}+2)$ migration rates is (see Fig. 6): $\{R_i, 0 \leq i \leq I_{MAX}+1\}$. Let Q be the integer-valued number of the pre-copy migration rates we select to update and let $S \equiv \frac{I_{MAX}}{Q}$ be the resulting integer-valued size of the rate clusters (see Fig. 6). Formally speaking, Q and S are to be selected according to the following two rules:

i) if $I_{MAX} \geq 1$, Q must be integer-valued and falling into the interval: $1 \leq Q \leq I_{MAX}$. Furthermore, Q must be selected so that the resulting ratio: $S \equiv \frac{I_{MAX}}{Q}$ is also integer-valued;

ii) if $I_{MAX} = 0$, the set $\{R_1, R_2, \ldots, R_{I_{MAX}}\}$ of the pre-copy rates is the empty one, so that we must pose: $Q = 1$ and $S = 0$.

Under these settings, the proposed SCBM is formally defined as follows:

i) the SCBM updates the following set of $(Q+2)$ migration rates: $\Xi \triangleq \{R_0; R_{jS+1},\ j = 0, 1, \ldots, (Q-1);\ R_{I_{MAX}+1}\}$;

ii) the SCBM sets the remaining $(I_{MAX} - Q)$ migration rates as follows: $R_i \equiv R_{jS+1}, \forall\, i = (jS+2), \ldots, (j+1)S$, and $\forall\, j = 0, 1, 2, \ldots, (Q-1)$.

Doing so, the expression in Eq. (2) for the downtime becomes:

$$\begin{aligned} T_{DT} = & M_0 \left(\overline{w}\right)^{(1+I_{MAX})} \left(\frac{1}{R_0 R_{I_{MAX}+1}}\right) \\ & \times \left[\prod_{k=0}^{Q-1} \left(\frac{1}{R_{kS+1}}\right)^{S}\right]. \end{aligned} \tag{25}$$

Furthermore, the resulting total migration time T_{TM} of Eq. (6) assumes, under the SCBM, the closed-form expression in (26) at the top of next page. As a consequence, the expression in Eq. (17) for the total energy $\mathcal{E}_{tot}$ wasted by the SCBM reads as in (27) at the top of next page.

Interestingly, the reported expressions in (25), (26) and (27) for the SCBM depend only on Q and the $(Q+2)$ migration rates to be updated (see the above defined set Ξ).

A. THE AFFORDED QOS MINIMUM-ENERGY BANDWIDTH OPTIMIZATION PROBLEM

In order to properly account for the metrics commonly considered for measuring the QoS of live VM migrations, we introduce four sets of hard constraints in the formulation of the minimum-energy SCBM optimization problem.

The first two constraints enforce QoS-dictated upper bounds: Δ_{MT} (s) and Δ_{DT} (s) on the tolerated migration

$$\begin{aligned} T_{TM} = M_0 \Biggl\{ \frac{1}{R_0} + (\overline{w})^{1+I_{MAX}} \left(R_0 R_{I_{MAX}+1}\right)^{-1} \left[\prod_{m=0}^{Q-1} \left(R_{mS+1}\right)^{-S} \right] \\ + \left(1-\delta\left(I_{MAX}\right)\right) \frac{1}{R_0} \Biggl\{ \sum_{k=0}^{Q-1} \sum_{l=kS+1}^{(k+1)S} (\overline{w})^{l} \Biggl\{ \delta(k) \left(R_1\right)^{-l} + (1-\delta(k)) \left[\prod_{p=0}^{k-1} \left(R_{pS+1}\right)^{-S} \right] \left(R_{kS+1}\right)^{-l+kS} \Biggr\} \Biggr\} \Biggr\}. \end{aligned} \tag{26}$$

$$\begin{aligned} \mathcal{E}_{tot} = \mathcal{E}_{setup} + \mathcal{P} T_{MT} \equiv \mathcal{E}_{setup} + K_0 M_0 \left(R_0\right)^{\alpha-1} + K_0 M_0 \left(R_0\right)^{-1} \Biggl\{ (\overline{w})^{1+I_{MAX}} \left(R_{I_{MAX}+1}\right)^{\alpha-1} \times \left[\prod_{k=0}^{Q-1} \left(R_{kS+1}\right)^{-S} \right] \\ + \left(1-\delta\left(I_{MAX}\right)\right) \Biggl\{ \sum_{m=0}^{Q-1} \Biggl\{ \sum_{l=mS+1}^{(m+1)S} (\overline{w})^{l} \Biggl(\delta(m) \left(R_1\right)^{\alpha-l} + (1-\delta(m)) \left[\prod_{p=0}^{m-1} \left(R_{pS+1}\right)^{-S} \right] \left(R_{mS+1}\right)^{\alpha+mS-l} \Biggr) \Biggr\} \Biggr\} \Biggr\}. \end{aligned} \tag{27}$$

time T_{MT} and downtime T_{DT}, respectively. Hence, they read as follows:

$$\varphi_1 \triangleq \frac{T_{MT}}{\Delta_{MT}} - 1 \leq 0, \tag{28}$$

and

$$\varphi_2 \triangleq \frac{T_{DT}}{\Delta_{DT}} - 1 \leq 0, \tag{29}$$

where T_{MT} and T_{DT} depend on the per-round aggregated migration rates of the underlying (equal-balanced) MPTCP connection (see Eqs. (25) and (26)).

The third constraint arises from the consideration that, in principle, the pre-copy stage in Fig. 3 of the PeCM technique could run indefinitely, if suitable stop conditions are not imposed. Commonly considered stop conditions account for [9]: (i) the maximum number I_{MAX} of the allowed migration rounds of Fig. 4; and, (ii) the maximum tolerated value: $\beta \geq 1$ of the ratio: (V_i/V_{i+1}) of the migrated data over two consecutive rounds. Hence, since the above constraints in Eqs. (28) and (29) already account for I_{MAX} through the corresponding expressions of T_{MT} and T_{DT} in Eqs. (25) and (26), we leverage Eq. (4), in order to formulate the constraints on the ratio: (V_i/V_{i+1}) in the following form:

$$\varphi_3\left(R_i\right) \triangleq \frac{\beta\overline{w}}{R_i} - 1 \leq 0, \tag{30}$$

for $i = 0$, and $i = jS+1$, $\forall\, j = 0, \ldots, (Q-1)$, where R_i denotes the aggregated bandwidth done available by the considered MPTCP connection during the i-th migration round of Fig. 3.

The last constraint accounts for the maximum uplink bandwidth that the 5G Network Processor of Fig. 1 allocates to the requiring device for the VM migration. In principle, depending on the bandwidth allocation policy actually implemented by the 5G Network Processor, the bandwidth assigned to the device could be exclusively allotted for the migration process (e.g., out-band migration) or it could be shared with the migrating application through statistical multiplexing (in-band migration; see [15]). In the first case, the bandwidth costraint reads as in: $R \leq R_{MAX}$, whilst, in the second case, we have: $R \leq \rho_{mgr} R_{AGR}$, where: (i) R_{MAX} (Mb/s) is the uplink bandwidth that the 5G Network Processor of Fig. 1 reserves for the exclusive support of the migration process; (ii) R_{AGR} (Mb/s) is the total aggregate bandwidth allocated by the 5G Network Processor for both the migration of the VM and the support of the migrating application; and, (iii) $\rho_{mgr} \in [0,\, 1]$ is the (dimensionless) fraction of the aggregate bandwidth R_{AGR} that is reserved by the wireless device for the migration process. Hence, after introducing the auxiliary position:

$$\widehat{R} \triangleq \min\left\{R_{MAX};\ \rho_{mgr} R_{AGR}\right\}, \tag{31}$$

the following set of bandwidth constraints:

$$R_i \leq \widehat{R}, \tag{32}$$

for $i = 0$, and $i = jS+1$, $\forall\, j = 0, \ldots, (Q-1)$, apply both cases of in-band (e.g., $R_{MAX} = \infty$) and out-band (e.g., $\rho_{mgr} R_{AGR} = \infty$) migration over the 5G FOGRAN of Fig. 1.

Overall, as a matter of these considerations, the afforded SCBM constrained optimization problem is formally stated as follows:

$$\begin{gathered} \min_{\substack{\{R_0, R_{jS+1}, R_{I_{MAX}+1}\} \\ j=0,1,\ldots,(Q-1)}} \mathcal{E}_{tot}, \\ \text{s.t. : Eqs. (28), (29), (30), and (32),} \end{gathered} \tag{33}$$

with $\mathcal{E}_{tot}$ given by Eq. (27) under the considered MPTCP-supported SCBM.

In the last part of this section, we discuss some possible generalizations and refinements of the stated SCBM optimization problem.

1) Accounting for wireless connection failures

According to the time chart of Fig. 3 and the related text, PeCM guarantees, by design, that the migrating VM continues to run on the wireless device till the end of the Commitment phase, so that the possible occurrence of failures of the device-to-fog MPTCP connection *do not* cause service interruption [9]. This inherent robustness of the PeCM technique against connection failure events makes it the preferred candidate for performing VM migration in failure-prone wireless environments [5]. However, nothing is for free, so that connection failure events still waste energy resources of the wireless device. In order to formally quantify the (average) energy loss caused by failures of the used MPTCP connection, we need to characterize the connection failure probability: $Pr_{cf}\left(\Delta_{MT}\right)$, formally defined as the probability that a connection failure event occurs during the migration interval, e.g., over the time: $[0, \Delta_{MT}]$. In general, in the considered mobile Fog scenario of Fig. 1, this probability may depend of a (large) number of (possibly, unpredictable) factors, such as, mobility speed and mobility trajectory of the device, radio coverage radius, utilized inter-cell handover mechanism, statistics of the wireless fading and so on, just to cite a few. However, the formal general analysis carried out in [36] supports the conclusion that the failure probability of a mobile connection is typically well described by the following Pareto-like expression:

$$Pr_{cf}(\Delta_{MT}) = \begin{cases} 1, & \text{for } \Delta_{MT} > T_{con}, \\ (1+\sigma)\left(\frac{\Delta_{MT}}{T_{con}}\right) - \sigma\left(\frac{\Delta_{MT}}{T_{con}}\right)^3, & \text{for } \Delta_{MT} \leq T_{con}. \end{cases} \tag{34}$$

In the above relationship, we have that: (i) σ is a dimensionless non-negative shaping factor, that accounts for the handover and/or mobility-induced heavy-tailed behavior of the connection failure probability; and, (ii) T_{con} (s) is the maximum expected duration of an on-going MPTCP connection. Although it may be hard to develop general computing formulae for these parameters, we note that, in our framework, they may be profiled on-line by the 5G Network Processor of Fig. 1, which typically records the statistics of the sustained connections [8]. Therefore, after profiling the connection failure probability and under the worst-case assumption that all the already migrated data are lost when a connection failure event happens, the resulting failure-induced energy loss suffered by the wireless device reads as the following product:

$$\overline{N}_{mgr} Pr_{cf}\left(\Delta_{MMT}\right) \overline{\mathcal{E}}_{tot}, \tag{35}$$

where $\overline{N}_{mgr} \geq 0$ is the (possibly, profiled) average number of VM migrations attempted by the wireless device of Fig. 1 during the time interval Δ_{MT}, and $\overline{\mathcal{E}}_{tot}$ is the average per-migration energy consumed by the wireless device.

2) Accounting for time-varying dirty rates of the migrating application

Let us consider the case in which the migrating application performs so many memory writing operations that the resulting dirty-rate changes during the migration time. Hence, after indicating by: $\overline{w}_i$ (Mb/s), and $w_{max} \triangleq \max_{1 \leq i \leq I_{MAX}+1} \left(\overline{w}_i\right)$ the dirty rate over the i-th migration round and the resulting maximum dirty rate respectively, let us introduce the following dummy positions: $\Gamma_i \triangleq \prod_{k=1}^{i} \overline{w}_i$, for $i \geq 1$, and $\Gamma_0 \triangleq 0$. Hence, after replacing: $\left(\overline{w}/R\right)^i$, $\left(\overline{w}/R\right)^{I_{MAX}+1}$ and $\overline{w}$ by: $\Gamma_i / \left(R\right)^i$, $\Gamma_i / \left(R\right)^{I_{MAX}+1}$ and w_{max} respectively, the resulting formulation of the considered bandwidth optimization problem directly applies to the case of time-varying dirty rates.

3) Accounting for the stretching of the execution times and memory compression of the migrating VM

During the execution of in-band VM migrations, the (previously introduced) aggregated bandwidth R_{AGR} available at the wireless device of Fig. 1 is partially utilized for migration purpose, so that the execution speed of the corresponding migrating application may decrease due to bandwidth contention phenomena [37]. In order to quantify the resulting stretching of the execution time of the migrating application, let $\overline{T}_{ex-mgr}$ and $\overline{T}_{ex}$ be the (average) execution times of the considered application in the presence/absence of VM migration, respectively. Hence, the queue analysis reported in [37] leads to the conclusion that the execution time stretching ratio: $\overline{T}_{ex-mgr}/\overline{T}_{ex}$ scales as

$$\frac{\overline{T}_{ex-mgr}}{\overline{T}_{ex}} = \frac{1}{1-\rho_{mgr}}, \tag{36}$$

where ρ_{mgr} is the (previously defined) fraction of the overall aggregated bandwidth that is reserved by the device for the VM migration. The above stretching ratio may be lowered by reducing the size of the migrated VM through compression coding (e.g., delta coding, run-length coding and similar). Hence, in order to simultaneously account for the contrasting effects on ρ_{mgr} of both compression coding and ARQ/FEC-based error-protection coding, let $\widetilde{M}_0$ (Mb) be the size of the uncompressed and uncoded VM, and let: $cp \in [0, 1]$, and $cr \in [0, 1]$ be the compression ratio and the overall coding rate of the ARQ/FEC-based error protection mechanisms implemented at both the MAC and Physical layers of the wireless device. Hence, the actual size M_0 of the VM to be migrated is directly computed as follows:

$$M_0 = \frac{cp}{cr}\widetilde{M}_0. \tag{37}$$

VII. FEASIBILITY CONDITIONS AND OPTIMIZED TUNING OF THE MAXIMUM NUMBER OF MIGRATION ROUNDS

Regarding the feasibility conditions of the stated bandwidth optimization problem, we observe that the involved constraint functions φ_1 and φ_2 in Eqs. (28) and (29) strictly decrease for increasing value of the aggregate transport rate R. Therefore, as a quite direct consequence, the following formal result holds.

Proposition 1: The SCBM optimization problem in Eq. (33) is feasible if and only if the following three conditions are simultaneously met:

$$\frac{M_0}{\Delta_{TM}}\left[\frac{1}{\widehat{R}}\left(I_{MAX}+2\right)\delta\left(\frac{\overline{w}}{\widehat{R}}-1\right) + \left(\frac{1-\left(\overline{w}/\widehat{R}\right)^{I_{MAX}+2}}{\widehat{R}-\overline{w}}\right)\left(1-\delta\left(\frac{\overline{w}}{\widehat{R}}-1\right)\right)\right] \leq 1, \tag{38}$$

$$\frac{M_0}{\Delta_{DT}}\frac{1}{\widehat{R}}\left(\frac{\overline{w}}{\widehat{R}}\right)^{I_{MAX}+1} \leq 1, \tag{39}$$

and:

$$\beta\left(\frac{\overline{w}}{\widehat{R}}\right) \leq 1. \tag{40}$$

Interestingly, the above feasibility conditions lead to some first insights about the effects of the parameters of I_{MAX}, $\widehat{R}$ and $\overline{w}$ on the (expected) behavior of the resulting SCBM. At this regard, three main remarks are in order. First, at fixed I_{MAX}, the expressions at the l.h.s. of Eqs. (38), (39) and (40) strictly increase (resp., decrease) for increasing values of the dirty rate $\overline{w}$ (resp., the maximum transport rate $\widehat{R}$). Second, for increasing values of I_{MAX}, the function at the l.h.s. of Eq. (39): (i) remains unchanged; (ii) decreases; and, (iii) increase, at $\left(\overline{w}/\widehat{R}\right)=1$, $\left(\overline{w}/\widehat{R}\right)<1$, and $\left(\overline{w}/\widehat{R}\right)>1$, respectively. Third, for increasing values of I_{MAX}, the l.h.s. of Eq. (38) increases, regardless from the value assumed by the ratio $\overline{w}/\widehat{R}$. The consequences are that: (i) increasing values of $\widehat{R}$ are welcome, because they always reduce both the resulting total migration time and downtime; but, (ii) increasing values of I_{MAX} lower the resulting downtimes at $\left(\overline{w}/\widehat{R}\right)<1$, whilst increase them at $\left(\overline{w}/\widehat{R}\right)>1$.

These considerations open the doors to the question regarding the optimized setting of I_{MAX}. Unfortunately, this is still an *open* question even in the case of state-of-the-art hypervisors, whose typically adopted application-oblivious default setting [9], [12]: $I_{MAX}=29$ does *not* guarantee, indeed, the convergence of the iterative pre-copy migration process and does *not* assure the minimization of the overall consumed energy (see, for example, [5], [38] and references therein). However, we anticipate that the carried out tests support the conclusion that, under the following setting of I_{MAX}:

$$\widetilde{I}_{MAX} \equiv \left\lceil \frac{\log\left(\frac{M_0}{\Delta_{DT}\widehat{R}}\right)}{\log\left(\frac{\widehat{R}}{\overline{w}}\right)} - 1 \right\rceil, \quad \text{for} \quad \left(\overline{w}/\widehat{R}\right)<1, \tag{41}$$

the resulting total migration energy wasted by the proposed SCBM typically attains its minimum. Intuitively, this is due to the fact that the expression in (41) is obtained by calculating the value of I_{MAX} that meets the feasibility constraint in (39) with the strict equality. Hence, under the setting in (41), the maximum tolerated downtime Δ_{DT} is fully exploited by the SCBM. This leads to a reduction of the average transport rate $\overline{R} \triangleq E\{R\}$ utilized during the overall migration process, that lowers, in turn, the resulting total migration energy. On the basis of this consideration, in the sequel, we refer to $\widetilde{I}_{MAX}$ in (41) as the optimized setting of I_{MAX} under the proposed SCBM.

VIII. PROPOSED SETTABLE-COMPLEXITY BANDWIDTH MANAGER AND RELATED IMPLEMENTATION ASPECTS

From a formal point of view, the energy function $\mathcal{E}_{tot}$ in (27) is a superposition of power-fractional terms of the type $(1/R_i)^{m_i}$, which involve the rate variables. Hence, $\mathcal{E}_{tot}$ is *not* a convex function in the transmission rates to be optimized, so that the resulting optimization problem in (33) is *not* a convex optimization problem. However, since all the $(Q+2)$ rate variables to be optimized are, by design, non-negative, we may introduce the following log-transformations:

$$\widetilde{R}_i \triangleq \log(R_i), \tag{42}$$

Thus:

$$R_i = e^{\widetilde{R}_i}, \tag{43}$$

Furthermore, after collecting the $(Q+2)$ log-rates in (42) in the following $(Q+2)$-dimensional (column) vector:

$$\overrightarrow{R} \triangleq \left[\widetilde{R}_0,\, \widetilde{R}_1,\, \widetilde{R}_{S+1},\, \ldots,\, \widetilde{R}_{(Q-1)S+1},\, \widetilde{R}_{I_{MAX}+1}\right]^T, \tag{44}$$

we indicate as in Eqs. (45), (46) and (47) at the top of next page, the resulting expressions for the total energy, downtime and migration time that are obtained by introducing the positions in (43) into Eqs. (25), (26) and (27), respectively. Hence, after posing: $R_i = e^{\widetilde{R}_i}$ in the Eq. (33) and related constraints, it may be directly viewed by inspection that the resulting log-transformed problem is *now jointly convex* in the involved log-transformed transmission rates: $\left\{\widetilde{R}_i\right\}$[2]. Furthermore, under the assumption that the feasibility conditions in (38) – (40) are satisfied with the strict inequalities, the so-called Slater's qualification condition holds [39], so that the solution of the considered optimization problem may be computed by applying the Karush-Khun-Tucker (KKT) optimality conditions (see, for example, Chapter 4 of [39] for a detailed presentation of this formal topic). Finally, due its convex form, the (log-transformed) box constraints in (32) on the maximum aggregate bandwidth of the underlying MTCP connection may be managed as implicit ones by performing orthogonal projections [39].

According to these observations, let:

$$\overrightarrow{\lambda} \triangleq \left[\lambda_1, \lambda_2, \lambda_{30}, \lambda_{31}, \lambda_{3(S+1)}, \ldots, \lambda_{3((Q-1)S+1)}\right], \tag{48}$$

be the non-negative $(Q+3)$-dimensional vector of the Lagrange multipliers associated to the $(Q+3)$ log-transformed inequality constraints in (28) – (30). Hence, the Lagrangian

[2] Formally speaking, this is a consequence of the fact that each term: $e^{\mp b_i \widetilde{R}_i}$ is a convex function in $\widetilde{R}_i$ for any value of the coefficient b_i. This is the formal reason that motivates the introduction of the log-transformations in Eq. (42).

$$\begin{aligned}\mathcal{E}_{tot}\left(\overrightarrow{\widetilde{R}}\right) = {} & \mathcal{E}_{setup} + K_0 M_0\, e^{(\alpha-1)\widetilde{R}_0} + K_0 M_0\, e^{-\widetilde{R}_0} \Bigg\{ (\overline{w})^{I_{MAX}+1} \times e^{\left[(\alpha-1)\widetilde{R}_{I_{MAX}+1} - S\widetilde{R}_1 - (1-\delta(Q-1))S\left(\sum_{k=1}^{Q-1} \widetilde{R}_{kS+1}\right)\right]} \\ & + (1-\delta(I_{MAX}))\left\{ \sum_{m=0}^{Q-1} \left\{ \sum_{l=mS+1}^{(m+1)S} (\overline{w})^l \left\{ \delta(m) e^{(\alpha-l)\widetilde{R}_1} + (1-\delta(m))\; e^{\left[(\alpha+mS-l)\widetilde{R}_{mS+1} - S\widetilde{R}_1 - (1-\delta(m-1))\left(\sum_{p=1}^{m-1} \widetilde{R}_{pS+1}\right)\right]} \right\}\right\}\right\}\Bigg\},\end{aligned} \tag{45}$$

$$T_{DT}\left(\overrightarrow{\widetilde{R}}\right) = M_0\, e^{-\widetilde{R}_0}\, (\overline{w})^{1+I_{MAX}} \times e^{\left[-\widetilde{R}_{I_{MAX}+1} - SR_1 - (1-\delta(Q-1))S\left(\sum_{k=1}^{Q-1} \widetilde{R}_{kS+1}\right)\right]}, \tag{46}$$

$$\begin{aligned}T_{TM}\left(\overrightarrow{\widetilde{R}}\right) = {} & M_0\, e^{-\widetilde{R}_0} \Bigg\{ 1 + (\overline{w})^{1+I_{MAX}}\;\; e^{\left[-\widetilde{R}_{I_{MAX}+1} - S\widetilde{R}_1 - (1-\delta(Q-1))S\right]} \times e^{\left(\sum_{k=1}^{Q-1} \widetilde{R}_{kS+1}\right)} \\ & + (1-\delta(I_{MAX}))\left\{ \sum_{k=0}^{Q-1} \left\{ \sum_{l=kS+1}^{(k+1)S} (\overline{w})^l \left\{ \delta(k)\, e^{-l\widetilde{R}_1} + (1-\delta(k)) \times\; e^{\left[(kS-l)\widetilde{R}_{kS+1} - S\widetilde{R}_1 - (1-\delta(k-1))S\left(\sum_{p=1}^{k-1} \widetilde{R}_{pS+1}\right)\right]} \right\}\right\}\right\}\Bigg\},\end{aligned} \tag{47}$$

function: $\mathcal{L}\left(\overrightarrow{\widetilde{R}},\, \overrightarrow{\lambda}\right)$ of the resulting log-transformed convex optimization problem reads as in Eq. (49) at the top of next page. Then, the minimization to be carried out is the following one:

$$\max_{\overrightarrow{\lambda} \geq \overrightarrow{0}} \left\{ \min_{\widetilde{R}_i \leq \log \widehat{R}} \left\{ \mathcal{L}\left(\overrightarrow{\widetilde{R}},\, \overrightarrow{\lambda}\right)\right\}\right\}. \tag{50}$$

$\forall\, i = 0, (S+1), \ldots, (Q-1)S+1, (I_{MAX}+1)$. The latter may be, in turn, computed by performing the orthogonal projection on the box sets in (50) of the vector which solves the nonlinear algebraic system that is obtained by equating to zero the $(2Q+5)$-dimensional vector gradient of the Lagrangian function in (50) with respect to both the primal $\overrightarrow{\widetilde{R}}$ and dual $\overrightarrow{\lambda}$ variables in (44) and (48), that is:

$$\overrightarrow{\nabla}\mathcal{L}\left(\overrightarrow{\widetilde{R}},\, \overrightarrow{\lambda}\right) = \overrightarrow{0}_{(2Q+5)\times 1}. \tag{51}$$

In this sequel, we indicate by $\left\{\overrightarrow{\widetilde{R}}^*,\, \overrightarrow{\lambda}^*\right\}$ the $(2Q+5)$-dimensional projected solution of this algebraic equations system. Being nonlinear, it does not admit a closed-form solution. However, its solution may be iteratively computed through a suitable set of projected gradient-based primal-dual iterations. At this regard, we note that, as pointed out in [10], the primal-dual algorithm is an iterative procedure for solving convex optimization problems, which applies quasi-Newton methods for updating the primal-dual variables simultaneously, in order to move toward the saddle-point (50) of the underlying Lagrangian function at each iteration.

In the proposed framework, the $(2Q+5)$ scalar updatings to be carried out at the n-th iteration read as follows[3]:

$$\widetilde{R}_i^{(n+1)} = \left[\widetilde{R}_i^{(n)} - \omega_i^{(n)}\, \nabla_{\widetilde{R}_i}\mathcal{L}\left(\overrightarrow{\widetilde{R}}^{(n)},\, \overrightarrow{\lambda}^{(n)}\right)\right]^{\log\left(\widehat{R}\right)}, \tag{52}$$

$$\lambda_i^{(n+1)} = \left[\lambda_i^{(n)} + \xi_i^{(n)}\, \nabla_{\lambda_i}\mathcal{L}\left(\overrightarrow{\widetilde{R}}^{(n)},\, \overrightarrow{\lambda}^{(n)}\right)\right]_+, \tag{53}$$

and

$$\lambda_{3i}^{(n+1)} = \left[\lambda_{3i}^{(n)} + \Psi_{3i}^{(n)} \nabla_{\lambda_{3i}}\mathcal{L}\left(\overrightarrow{\widetilde{R}}^{(n)},\, \overrightarrow{\lambda}^{(n)}\right)\right]_+, \tag{54}$$

for $n \geq 0$, $\lambda_{3i}^{(0)} = 0$, and where $\left\{\omega_i^{(n)},\, n \geq 0\right\}$, $\left\{\xi_i^{(n)},\, n \geq 0\right\}$ and $\left\{\Psi_{3i}^{(n)},\, n \geq 0\right\}$ are $(2Q+5)$ suitable non-negative sequences of n-varying step-sizes. The projections in (52), (53) and (54) account for the box-type constraints in (50).

A. ADAPTIVE IMPLEMENTATION AND SETTABLE IMPLEMENTATION COMPLEXITY OF THE PROPOSED SCBM

It is expected that, due to fast fading phenomena and/or device mobility, the state of the MPTCP connection of Fig. 1 may experience abrupt time fluctuations during the migration process [36], [40]. In order to effectively cope with these (typically unpredictable) fluctuations, the gain sequences in Eqs. (52), (53) and (54) must be adaptively updated, in order to allow the SCBM to quickly response in a reactive way, e.g., without the support of (typically unreliable and error-prone) forecasting of the fading levels and/or device mobility platforms.

At this regard, we note that the Lyapunov-based formal analysis of the stability of MPTCP connections recently

[3] By definition, $[x]^{\alpha}$ means: $\min\{x;\, \alpha\}$, while $[x]_+$ denotes: $\max\{x;\, 0\}$.

$$\begin{aligned}\mathcal{L}\left(\overrightarrow{\overrightarrow{R}},\, \overrightarrow{\lambda}\right) \triangleq \mathcal{E}_{tot}\left(\overrightarrow{\overrightarrow{R}}\right) + \lambda_1 \left\{\left[\frac{1}{\Delta_{TM}}\, T_{TM}\left(\overrightarrow{\overrightarrow{R}}\right)\right] - 1\right\} + \lambda_2 \left\{\left[\frac{1}{\Delta_{DT}} T_{DT}\left(\overrightarrow{\overrightarrow{R}}\right)\right] - 1\right\} \\ + \left\{\lambda_{30}\left[\beta\, \overline{w}\, e^{-\widetilde{R}_0} - 1\right] + \left\{\sum_{m=0}^{Q-1} \lambda_{3(mS+1)} \left[\beta\, \overline{w}\, e^{-\widetilde{R}_{Sm+1}} - 1\right]\right\}\right\}\end{aligned} \tag{49}$$

reported in [10] guarantees that the asymptotical global stability of the MPTCP CC algorithms of Section IV is attained when each gain sequence: $\left\{g_y^{(n)},\, n \geq 1\right\}$ required to update the corresponding sequence: $\left\{y^{(n)},\, n \geq 1\right\}$ of the associated primal/dual variable is computed step-by-step on the basis of the following (memoryless and uncoupled) relationship (see Theorem 3.3 of [10] for a formal proof of this result):

$$g_y^{(n)} = \frac{1}{2}\left(y^{(n)}\right)^2. \tag{55}$$

The scenario considered in [10] is of wired-type, so that the attainment of quick responsiveness is typically not a major concern. However, in the considered wireless/mobile environment of Fig. 1, the following two additional goals should be, indeed, also attained. First, $g_y^{(n)}$ must remain sufficiently large for vanishing $y^{(n)}$, in order to allow the SCBM to fast react to abrupt environmental changes. Second, $g_y^{(n)}$ must not grow too much for increasing values of $y^{(n)}$, in order to limit the occurrence of unstable oscillatory phenomena. Overall, motivated by these considerations, we propose to update the gain sequences in Eqs. (52), (53) and (54) according to the following "clipped" n-indexed relationships:

$$\omega_i^{(n)} = \max\left\{\frac{a_{MAX}}{10};\, \min\left\{a_{MAX};\, \frac{1}{2}\left(\widetilde{R}_i^{(n)}\right)^2\right\}\right\}, \tag{56}$$

$$\xi_i^{(n)} = \max\left\{\frac{a_{MAX}}{10};\, \min\left\{a_{MAX};\, \frac{1}{2}\left(\lambda_i^{(n)}\right)^2\right\}\right\}, \tag{57}$$

and:

$$\Psi_{3i}^{(n)} = \max\left\{\frac{a_{MAX}}{10};\, \min\left\{a_{MAX};\, \frac{1}{2}\left(\lambda_{3i}^{(n)}\right)^2\right\}\right\}, \tag{58}$$

In the above relationships, a_{MAX} is a positive parameter, whose goal is to clip from below and above the instantaneous values assumed by the gain sequences, in order to assure: (i) fast responsiveness in the transient-state; (ii) fast convergence to the steady-state, and, (iii) stability in the steady-state. At this regard, we anticipate that the carried out numerical tests of Section IX show that the performances of the resulting adaptive SCBM remain nearly unchanged over a quite broad range of values of a_{MAX}. This supports, in turn, the conclusion that the behavior of the adaptive SCBM is not too much sensitive with respect to the actual setting of a_{MAX}.

1) Implementation complexity aspects

From an implementation point of view, the primal-dual iterations in Eqs. (52), (53) and (54) and the associated ones in Eqs. (56), (57) and (58) must be carried out by the wireless device of Fig. 1 at the end of the Pre-migration phase of Fig. 3. Hence, the duration T_{IT} (s) of each n-indexed iteration should be small enough to allow the convergence to the (global) optimum within a small portion of the Pre-migration phase. At this regard, we have numerically ascertained that it suffices to set T_{IT} to twenty times the inverse of the maximum clock frequency of the device CPU.

Passing to consider the implementation complexity of the adaptive SCBM, an inspection of Eqs. (52), (53) and (54) points out that it is *independent* from the allowed maximum number of round I_{MAX}, and it is affected by two factors, namely, the number of gradient iterations to be updated at each step n and the number of the n-indexed steps needed for their convergence. Since the number of iterations equates (see Eqs. (52) – (54) and (56) – (58)):

$$2 \times (Q + 5)\,, \tag{59}$$

we conclude that the dependence of the implementation complexity of the adaptive SCBM on the (settable) number Q of migration rates is of *linear* type. At this regard, we anticipate that the carried out numerical tests show that values of Q as small as 2 – 3 suffice for attaining the minimum migration energy. The implementation complexity also scales up, by design, in a *linear* way with the number of the n-indexed iterations requested for the convergence of Eqs. (52) – (54). Although this number may depend, in turn, on the selected value of Q, we have numerically ascertained that, in general, the clipping action of the a_{MAX} parameter in (56) – (58) allows to attain small convergence times of the order of about 5 – 6 n-indexed iterations (see Section IX).

2) Real-time profiling of the application dirty-rate and MPTCP connection

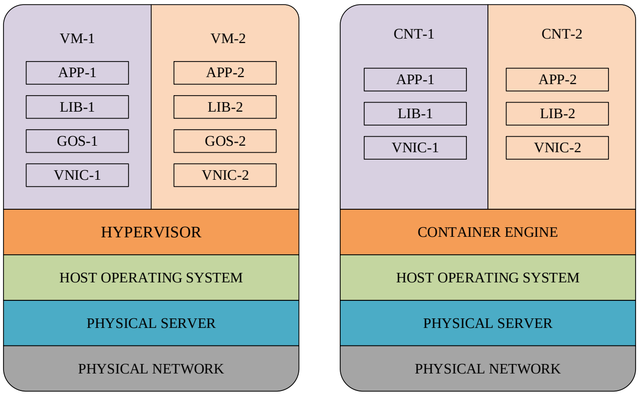
The memory size M_0 of the migrating VM may be measured during the Pre-migration phase of Fig. 3 by issuing the *xenstore* command of the legacy Xen hypervisor [12]. In order to profile on-line the average dirty rate $\overline{w}$ of the migrating application, we point out that state-of-the-art hypervisors (such as, for example, the Xen hypervisor) periodically report the map of the changed (e.g., dirtied) memory cells of the migrating VM, in the form of dirty bitmaps [12]. Hence, the actual dirty rate $\overline{w}$ of the migrating VM may be profiled on-line by averaging over the migration time the number of memory cells marked as dirtied in the reported bitmaps.

Passing to consider the online power-profiling of the (equal-balanced) MPTCP connection of Section IV-D, we begin to observe that the subflow round-trip-times and the corresponding maximum available bandwidths may be directly profiled at the Transport layer of Fig. 5 by issuing, for example, the *iperf* command of Linux [12]. In order to profile online the setup energy in (12) and the K_0 and α parameters in (17), the OS of the wireless device may issue the *ifconfig* command of Xen on a per-WNIC basis, in order to acquire the actual power states of the utilized WNICs [12]. Specifically, under the equal-balanced condition of Eq. (16), by issuing the *ifconfig* command at vanishing and unit-value transport rate R, the device directly measures $\mathcal{E}_{setup}$ in (12) and K_0 in (17), respectively. Afterward, by issuing once a time the *ifconfig* command at $R = \widehat{R}$, the device measures the total static-plus-dynamic power $\mathcal{P}_{tot}\left(\widehat{R}\right)$ consumed by the underlying WNICs. Finally, since the corresponding dynamic power $\mathcal{P}\left(\widehat{R}\right)$ equates, by definition, the difference: $\mathcal{P}_{tot}\left(\widehat{R}\right) - \mathcal{E}_{setup}\widehat{R}$, by solving Eq. (17) with respect to α, we obtain the following formula for the online profile of the α exponent:

$$\alpha = \frac{\log\left(\left(\mathcal{P}_{tot}\left(\widehat{R}\right) - \mathcal{E}_{setup}\widehat{R}\right)/K_0\right)}{\log \widehat{R}}. \tag{60}$$

3) Virtualization-vs.-Containerazation technologies

The basic feature of the virtualized Fog node of Fig. 1 is that each served wireless device is mapped into a virtual clone that acts as a virtual processor and executes the migrated application on behalf of the wireless device. In addition to the here considered VM-based virtualization technology, the emerging CoNTainer (CNT)-based technology is quickly gaining momentum [41]. Figure 7 reports "win-to-win" the basic architectures of these two technologies.



(a) VM-based server virtualization

(b) Container-based server virtualization

FIGURE 7: VM-vs.-Container based server virtualization. VM = Virtual Machine; LIB = LIBrary; GOS = Guest Operating System; VNIC = Virtual Network Interface Card; CNT = CoNTainer.

Shortly, their main architectural differences are that [41]: (i) the VM technology relies on a middleware software layer (e.g., the so called Hypervisor), that statically performs hardware virtualization, while the CNT technology uses an Execution Engine, in order to dynamically carry out resource scaling and multiplexing; and, (ii) each VM is equipped with an own (typically, heavy-weight) Guest Operating System (GOS), while a container comprises only application-related (typically, light-weight) libraries, and shares with the other containers the Host Operating System (HOS) provided by the physical server. The resulting pros and cons of these two virtualization technologies are summarized in Table 7.

In a nutshell, the main pros of the CNT-based virtualization are that: (i) containers are light-weight and can be bootstrapped significantly quicker than VMs; and, (ii) the physical resources required by a container can be scaled up/down in real-time by the corresponding Execution Engine. However, since all containers hosted by a same physical server share the same HOS, the main cons of the CNT-based virtualization are that: (i) the level of resource isolation offered by the CNT-based virtualization is lower than the corresponding one of the VM-based technology; and, (ii) at the present time, the CNT technology does not support *live* migration (see the second row of Table 7). This the reason why in this contribution we directly refer to the VM technology.

At this regard, we remark that the so-called Checkpoint/Restore in Userspace (CRIU) technique is today utilized as tool for the migration of containers [42]. However, the CRIU technique is of Stop-and-Copy type, so that it is not suitable for the migration of delay-sensitive real-time applications [41]. Motivated by this consideration, container developers are currently working on a new project[4], in order to develop a middleawre platform for the support of live migration of containers. Since the proposed SCBM does not exploit, by design, primitives that are specific of the VM virtualization technology, we expect that it will be able to run unchanged even atop future middleware layers that support live migration of containers.

IX. PERFORMANCE TESTS

The simulated scenario is the 5G FOGRAN of Fig. 1. However, due to its multi-facet nature, it is (very) challenging to build up a testbed and, then, carry out repeatable field trials. Hence, motivated by this consideration, we chose numerical simulations as a means to evaluate the performance of the proposed adaptive SCBM, in order to assure both the repeatability of the performed tests and carry out fair performance comparisons. The MATLAB toolkit is used as the simulation platform, mainly because it natively supports a number of primitives for the numerical implementation of optimization solvers. The simulated scenario refers to the aforementioned in-band PeCM VM migration running atop the end-to-end virtualized architecture of Fig. 2.

In detail, the simulated platform emulates the wireless interconnection between an iPhone7-Plus smartphone and a

[4] https://criu.org/P.Hau

TABLE 7: Comparisons of the main native features of the VM and Container virtualization technologies.

Attributes	Virtual Machine	Container
Guest OS	Each VM is equipped with an own Guest OS that runs atop the hardware abstraction provided by an underlying hypervisor	All the containers share a same OS that is directly stored by the physical memory of the hosting server
Migration support	Middleware support is provided for live migration of VMs	Live container migration is not provided. Only Stop-and-Copy type migration is currently supported
Virtualization-induced slowdown	Due to the presence of both the Guest and Host operating systems, VMs suffer from virtualization-induced slowdowns	Containers offer quasi native computing speeds as compared to the underlying Host OS
Resource isolation	Quasi perfect isolation between VMs running on a same server is attained by forbidding the sharing of files and libraries	Inter-container file sharing is allowed that reduces resource isolation
Bootstrap delay	The bootstrap delay of a VM is of the order of some minutes	The bootstrap delay of a Container is of the order of some seconds
Storage requirement	VMs demand for not negligible storage space, in order to install the whole Guest OS and associated libraries	Containers require less storage space than VMs, because the base OS is shared

virtualized Dell Power Edge server that acts as a Fog node. The smartphone is assumed to be equipped with the iOS7 operating system, that already supports MPTCP for enabling its SIRI-based voice recognition applications[5]. The smartphone is assumed to be also equipped with: (i) 1.4 GHz CPU and 128 MB RAN; (ii) IEEE802.11b, 3G-UTRA and 4G-LTE WNICs; and, (iii) a Xen 3.3-like hypervisor [12], that provides the basic virtualization functions and manages them according to the reference architecture of Fig. 2. The Dell Power Edge server at the simulated Fog node is assumed to be equipped with 3.06 GHz Intel Xeon CPU and 4GB RAM, and its hardware resources are virtualized by a Xen-compliant hypervisor. WiFi connectivity is assumed to be guaranteed by a WRT54GL-type access point that is co-located with the server, while national-wide cellular operators provide mobile 3G/4G cellular coverage. The proposed adaptive SCBM is assumed to be implemented in software at the driver domain (e.g., Dom-0; see [12]) of the (aforementioned) legacy Xen 3.3 hypervisor. Furthermore, in order to avoid performance interferences, we also assume that the driver domain is forced to use a single physical core of the wireless device, whilst the migrating VM runs atop a second physical core. The device-to-fog simulated wireless connection is impaired by frequency-flat Rice fading with Rice factor of 6.5, and the coverage radius of the WiFi link is 300 (m). Finally, in order to account for the signaling overhead introduced by the protocol stack of Fig. 5, the maximum rates available at the MPTCP layer are assumed to be limited up to 90% of the corresponding raw bit rates measured at the Physical layer.

Table 8 reports the values of the main simulated parameters, that are compliant with those typically considered by the open literature (see, for example, in [33], [35]). It is understood that the setup energies of Eq. (12) are evaluated by multiplying the corresponding setup powers in the last row of Table 8 by the considered total migration times Δ_{TM}.

A. BENCHMARK BANDWIDTH MANAGERS

The benchmark bandwidth managers considered in the following subsections for performance comparisons are those described in [9] and [15], that are also the only ones currently available in the open literature.

Specifically, on the basis of some (recent) surveys, as well as at the best of the authors' knowledge, the bandwidth manager originally proposed in [9] is the "de facto" standard one that currently equips a number of commercial hypervisors, like Xen, KVM and WMware (see [5] and references therein). Roughly speaking, it implements a heuristic best effort policy, in which the total migration bandwidth[6]: R^{XEN} is incremented from the minimum: $R_{MIN}^{XEN} \equiv \overline{w}$ to the maximum: $R_{MAX}^{XEN} \equiv \widehat{R}$ by passing from the 0-th initial migration round to the $\left(I_{MAX}^{XEN}+1\right)$-th final round (see Fig. 4). For this purpose, at the beginning of each round, the migration bandwidth is incremented by the quantity [9]: $\Delta R^{XEN} \equiv \left(R_{MAX}^{XEN}-\overline{w}\right) / \left(I_{MAX}^{XEN}+1\right)$. The rationale behind this heuristic approach is to shorten as much as possible the final downtime by allowing to migrate at the maximum transmission rate: $R_{MAX}^{XEN} \equiv \widehat{R}$ during the final Stop-and-Copy phase of the migration process (see Figs. 3 and 4). Doing in so, this heuristic does not consider at all the resulting network energy consumption induced by the

[5] https://support.apple.com/lv-lv/HT201373

[6] In the sequel, we denote by the upper-scripts: $(\cdot)^{XEN}$ and $(\cdot)^{LIV\_MIG}$ the performance metrics and working parameters of the considered benchmark bandwidth managers in [9], [15], respectively. The corresponding performance metrics and working parameters of the proposed adaptive SCBM are marked as: $(\cdot)^{SCBM}$.

TABLE 8: Main simulated parameters [33], [35].

Parameter	3G	4G	IEEE802.11b WiFi
α	2	2	2
MSS (Mb)	8×10^{-3}	8×10^{-3}	8×10^{-3}
R_{MAX} (Mb/s)	0.9×2	0.9×50	0.9×11
$\overline{RTT}$ (ms)	250	35	25
Ω (W$^{0.5}$)	10^{-4}	5.1×10^{-3}	10^{-2}
$\mathcal{P}_{setup}$ (mW)	87.83	137.83	159.46

migration process, and the numerical results reported in the sequel confirm, indeed, its energy sub-optimality.

In order to improve the energy performance of the bandwidth manager in [9], the recent contribution in [15] develops a formal approach that is based on the constrained optimization of the total energy: $\mathcal{E}_{tot}$ wasted by the migration process. However, unlike the here proposed SCBM, the bandwidth manager in [15] is designed for operating under wired intra-data center environments, and, then (see [15]): (i) it allows the optimization of *only* a *single* migration rate, e.g., it uses a single optimized transmission rate over all the performed migration rounds; (ii) it is not optimized for working under MPTCP, e.g., only the case of SPTCP is considered in [15]; and, (iii) its adaptation mechanism is optimized for attaining a stable behavior in the steady-state, so that its responsiveness is, indeed, quite limited. As a matter of these facts, the numerical results reported in the sequel point out that the energy performance of the bandwidth manager in [15] is no so good under the here considered FOGRAN mobile scenarios.

B. DESCRIPTION OF THE OBTAINED NUMERICAL RESULTS AND PERFORMANCE COMPARISONS

Goal of this subsection is threefold. First, we present a number of numerical results that aim at giving insights into the implementation complexity-vs.-energy performance tradeoff and the adaptive capability of the proposed SCBM. Second, we present some comparative results about the energy performance of the SCBM with respect to the (previously described) benchmark bandwidth managers in [9] and [15] under both synthetic and real-world workloads. Third, we compare the energy performances of the proposed SCBM under EWTCP MPTCP and newReno SPTCP. At this regard, we remark that a number of formal and measurement-based studies confirm that the enhanced aggregation bandwidth and responsiveness capabilities exhibited by the EWTCP CC algorithm make it (very) suitable for wireless/mobile applications (see, for example, [14], [19]–[21] and references therein). Therefore, unless otherwise stated, in the sequel, we focus on the numerical results for the case of equal-balanced EWTCP connections that operate at (see Eq. (17)) $K_0 = 2.5\times10^{-2}$ (W $\times$ (Mb/s)^2) by simultaneously utilizing the 3G/4G/WiFi transmission technologies featured in Table 8. However, we explicitly remark that, thanks to the unified power analysis of Section IV, the corresponding steady-state energy performance of the other MPTCP CC algorithms may be obtained by scaling the corresponding expressions of K_0 of Table 6.

1) Considered synthetic and real-world test applications

Both synthetic and real-world applications have been selected, in order to simulate the workloads actually generated by the migrating VMs. Specifically, we have used the *memtester* application for the generation of synthetic workloads[7]. This is a write-intensive application that it is currently used to detect faults in RAM by allowing the programmer to set the desired value of the average memory dirty rate $\overline{w}$. As a set of real-world benchmark applications, the *bzip2*, *mcf* and *memcached* programs have been selected from the SPEC CINT2000 benchmark tool [8]. The first one is a CPU and memory read-intensive application, that is featured by a quite low memory dirty rate. The second one is characterized by a larger dirty rate, because it presents a balanced mix of read and write memory operations. Finally, the third one is a write-intensive memory program, that emulates the rate-intensive caching of multiple key/value pairs in the RAM of the migrating applications.

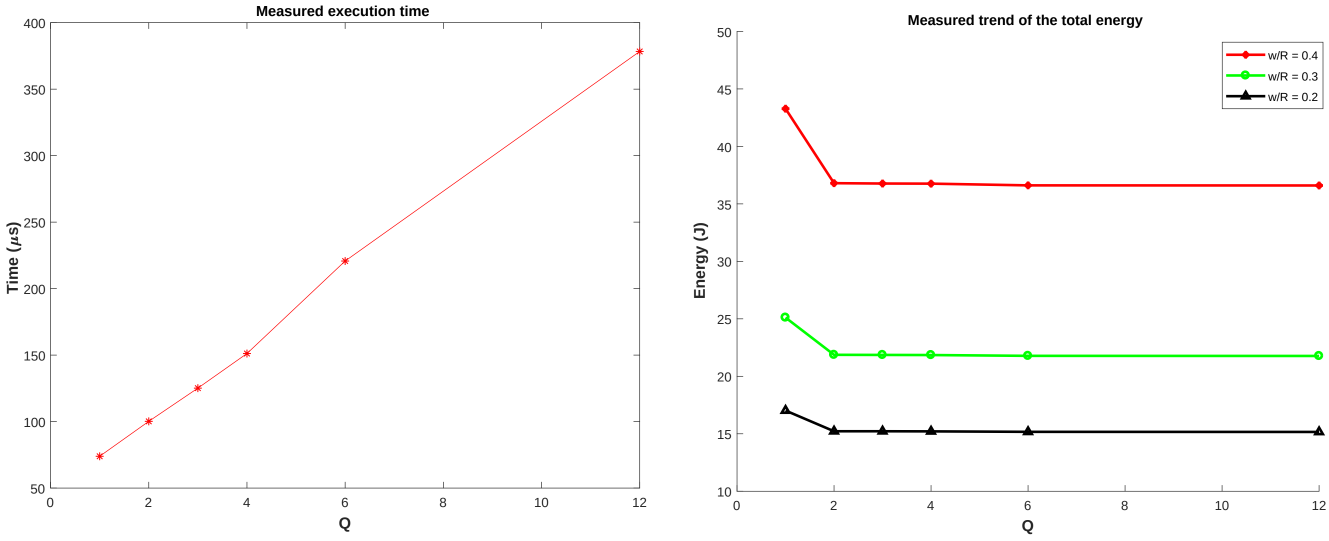
2) Implementation complexity-vs.-energy performance trade-off under the proposed SCBM

The numerical plots of Fig. 8 are obtained under the synthetic workloads generated by the *memtester* application. Their goal is to give some first insights about the interplay between settable implementation complexity and energy performance of the proposed bandwidth manager. Specifically, Fig. 8a reports the numerically measured execution times of the proposed SBCM for increasing values of the settable number Q of the migration rates to be optimized.

Interestingly, the measured trend of the execution times is nearly linear and, then, it is compliant with the implementation complexity formula of Eq. (59). Regarding the actual values of Q needed to drive the total energy $\mathcal{E}_{tot}^{SCBM}$ consumed by the proposed SCBM to its global minimum, the behaviors of the plots of Fig. 8b support two main conclusions. First, since the volume of the data to be migrated increases for increasing values of the ratio $\left(\overline{w}/\widehat{R}\right)$

[7] Memtester-Memory Diagnostic tool, available at: http://pyropus.ca/software/memtester

[8] SPEC CPU2000. Standard Performance Evaluation Corporation, available at: http://www.spec.org./cpu2000/CINT200



(a) (b)

FIGURE 8: (a) Measured execution times of the proposed SCBM for increasing values of Q. (b) Behaviors of the total energy $\mathcal{E}_{tot}^{SCBM}$ consumed by the proposed SCBM for increasing values of Q at $\left(\overline{w}/\widehat{R}\right) = 0.2, 0.3, 0.4$ under the considered EWTCP-based scenario.

(e.g., for increasing values of the dirty rate of the migrating application), the plots of Fig. 8b scale up for growing values of $\left(\overline{w}/\widehat{R}\right)$. Second, the semi-flat behaviors of the plots of Fig. 8b support the conclusion that small values of Q ranging from 1 to 3 are suffice, in order to drive the total energies consumed by the SCBM to their global minima. Since analogous trends have been, indeed, obtained under all the simulated scenarios, we may conclude that, in practice, the implementation complexity of the Q-optimized SCBM is limited and of the order of (see Eq. (59) with $Q = 1, 3$): $\mathcal{O}(12) - \mathcal{O}(16)$.

3) Responsiveness of the proposed SCBM

Goal of the plots of Figures 9 and 10 is to numerically check the actual responsiveness of the adaptive SCBM when, due to memory-contention phenomena and/or fluctuations of the round-trip-times of the utilized transport connections, the dirty rate of the migrated VM and/or the state of the EWTCP connection undergo unpredicted quick changes. At this regard, the plots of Figure 9 (resp., Figure 10) report the numerically evaluated trajectories of the n-indexed sequence: $\left\{\mathcal{E}_{tot}^{SCBM(n)}, n \geq 1\right\}$ of the total energy consumed by the SBCM when the initial value of the dirty rate: $\overline{w} = 1$ (Mb/s) of the migrated VM (resp., the initial value: $K_0 = 2.5 \times 10^{-2}$ (W $\times$ (Mb/s)^2) of the power-profile of the underlying EWTCP connection) jumps to: $2 \times \overline{w}$ (resp., to: $10 \times K_0$) at $n = 8$, and, then, comes back to its initial values at $n = 20$. The workload of the *memtester* application has been generated, in order to simulate the migrating VMs.

An examination of these plots leads to two main insights. First, the relationships in Eqs. (56) – (58) designed for updating the gain sequences of the gradient-based iterations of Eqs. (52) – (54) allow the SCBM to be very responsive, with convergence times that are typically limited up to 4 – 6 n-indexed iterations. Second, the sensitivity of these convergence times on the actual value assumed by the "clipping" parameter a_{MAX} in Eq. (56) – (58) is not substantial, at least for values of a_{MAX} falling into the (quite broad) interval: $3 \times 10^{-3} \div 3 \times 10^{-2}$.

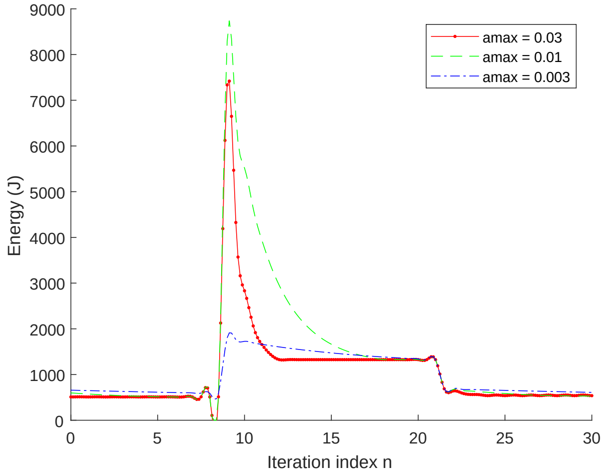


FIGURE 9: Time evolutions (in the n index) of the energy consumption of the adaptive SCBM under the considered EWTCP-based scenario. Case of time-varying dirty rate $\overline{w}$ at: $\widehat{R} = 18$ (Mb/s), $M_0 = 64$ (Mb), $\beta = 2$, $\Delta_{TM} = 50.5$ (s), and $\Delta_{DT} = 5.6$ (ms).

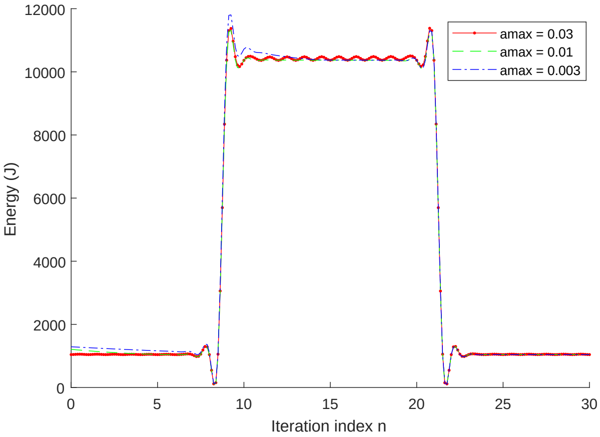


FIGURE 10: Time evolutions (in the n index) of the energy consumption of the adaptive SCBM under the considered EWTCP-based scenario. Case of time-varying K_0 of the EWTCP connection at: $\widehat{R} = 18$ (Mb/s), $M_0 = 64$ (Mb), $\beta = 2$, $\Delta_{TM} = 50.5$ (s), $\Delta_{DT} = 5.6$ (ms).

4) Performance comparisons under synthetic workloads

Goal of this subsection is to compare the energy performance of the proposed SCBM against the corresponding ones of the (previously described) benchmark bandwidth managers of Section IX-A, when the migrating VM is running the synthetic workload generated by the *memtester* application. The obtained numerical results are reported in Table 9, together with the corresponding *numerically optimized* values of I_{MAX}^{XEN}, $\widetilde{I}_{MAX}$ and Q.

An examination of this Table 9 leads to five main conclusions.

First, at assigned values of the downtime, migration time and speed up factor (see the first three rows of Table 9), we have numerically ascertained that the number of the maximum migration rounds that minimizes the energy consumptions of both the proposed SCBM and the benchmark one in [15] is given by the relationship of Eq. (41). Second, the corresponding (numerically evaluated) optimized values of the maximum number of migration rounds: I_{MAX}^{XEN} that allow the Xen manager to attain its minimum energy consumptions are quite *larger* than the corresponding ones: $\widetilde{I}_{MAX}$ of Eq. (41) (compare the 4-th and 5-th rows of Table 9). Third, as it could be expected, decreasing values of the tolerated downtimes lead to larger values of the energies consumed by all tested bandwidth managers (see the 7-th, 8-th and 9-th rows of Table 9). Forth, the per-cent energy savings: $\left(1 - \left(\mathcal{E}_{tot}^{SCBM}/\mathcal{E}_{tot}^{XEN}\right)\right)$ (%) of the proposed SCBM over the benchmark Xen one in [9] are over 25% and approach 45% under strict downtimes (see the 10-th row of Table 9). At this regard, we have numerically ascertained that about 25% – 30% of these attained energy savings are induced by the application-aware setting of $\widetilde{I}_{MAX}$ of Eq. (41), and this supports the actual effectiveness of this setting. Fifth, the corresponding per-cent energy savings: $\left(1 - \left(\mathcal{E}_{tot}^{SCBM}/\mathcal{E}_{tot}^{LIV\_MIG}\right)\right)$ (%) of the proposed SCBM over the benchmark bandwidth manager in [15] maintain around 12% (see the last row of Table 9). Since these last energy savings are obtained at $Q = 1$ (see the 6-th row of Table 9), we conclude that small increments in the implementation complexity of the proposed SCBM (e.g., small values of the settable Q parameter) over the benchmark manager in [15] suffice, in order to attain noticeable energy reductions.

5) Performance comparisons under real-world workloads

These conclusions are corroborated by the additional numerical results of this section. Specifically, its goal is to numerically test and compare the energy performance of the proposed SCBM under *hard* operating conditions featured by *large* sizes of the migrating VMs and *low* values of the tolerated downtimes. In order to (attempt to) to provide a comprehensive picture covering heterogeneous application scenarios, all three the (aforementioned) real-world workloads: *bzip2*, *mcf* and *memcached* have been considered in the carried out tests. The upper part of Table 10 reports the main parameters of the simulated real-world workloads, whilst the middle and bottom parts of this table point out both the energy consumptions of the considered bandwidth managers and the resulting per-cent energy savings of the proposed SCBM over the benchmarks ones. In all the carried out tests, the sizes of the migrated VMs scaled up to $M_0 = 128$ (Mb), whilst the tolerated downtimes scaled down to $\Delta_{DT} = 1.0$ (ms).

Overall, a synoptic examination of the numerical data of Table 10 leads to three main conclusions. First, it is confirmed that the energy consumptions of all the considered bandwidth managers increase for increasing values of the ratio $\left(\overline{w}/\widehat{R}\right)$. Second, under the here considered “harsh” operating conditions, the per-cent energy savings of the proposed SCBM over the benchmark XEN one (resp., the benchmark LIV_MIG one) maintain over 60% (resp., over 50%) and approach 70% (resp., 64%) for values of the ratio $\left(\overline{w}/\widehat{R}\right)$ of the order of 0.7. Third, in order to attain these (large) energy reductions, the Q parameter of the SCBM scaled up to: $Q = 3$.

6) SCBM atop EWTCP and newReno SPTCP: performance comparisons

MPTCP (and, then, EWTCP) utilizes multiple WNICs in parallel. Therefore, in principle, it could happen that the energy reductions provided by the bandwidth aggregation is offset by the increment of the static energy in Eq. (12) needed to simultaneously turning-ON the utilized WNICs, so that the resulting total energy consumed by MPTCP could be larger than the corresponding ones wasted by commodity SPTCP [14], [21]. However, it is also reasonable to expect that, in our framework, this energy offset may (eventually) happen when the constraints on the allowed downtimes are (very) relaxed, and/or the dirty rates and sizes of the migrating VMs are (very) low.

TABLE 9: Energy performances of the proposed SCBM and the considered benchmark bandwidth managers under the considered EWTCP-based scenario at: $M_0 = 128$ (Mb) and $\overline{w}/\widehat{R} = 0.33$. The (numerically evaluated) reported values of I_{MAX}^{XEN}, $\widetilde{I}_{MAX}$ and Q are the optimal ones, e.g., they minimize the energy consumptions of the corresponding bandwidth managers.

Δ_{DT} (s)	0.103	5.42×10^{-4}	4.03×10^{-5}
Δ_{TM} (s)	46.9	65.2	83.6
β	2	2	2
I_{MAX}^{XEN}	6	14	29
$\widetilde{I}_{MAX}$	4	9	11
Q	1	1	1
$\mathcal{E}_{tot}^{XEN}$ (J)	1880	2150	2470
$\mathcal{E}_{tot}^{LIV\_MIG}$ (J)	1550	1557	1560
$\mathcal{E}_{tot}^{SCBM}$ (J)	1366	1373	1373
Energy saving over XEN (%)	27.3	36.1	44.4
Energy saving over LIV_MIG (%)	11.8	11.8	11.9

TABLE 10: Simulated parameters, obtained energy consumptions and resulting per-cent energy savings under the considered EWTCP-based scenario for the real-world workloads *bzip2*, *mcf* and *memcached* at: $\beta = 2$, $\Delta_{TM} = 81.5$ (s), and $\Delta_{DT} = 1.0$ (ms). The reported results of the SCBM refer to $Q = 3$.

Parameter	***bzip2***	***mcf***	***memcached***
M_0 (Mb)	128	128	128
$\overline{w}$ (Mb/s)	0.72	1.08	1.26
$\overline{w}/\widehat{R}$	0.4	0.6	0.7
$\mathcal{E}_{tot}^{XEN}$ (J)	107	196	228
$\mathcal{E}_{tot}^{LIV\_MIG}$ (J)	100	128	135
$\mathcal{E}_{tot}^{SCBM}$ (J)	39.4	60.34	66.4
Energy saving respect XEN (%)	60.50	69.21	70.84
Energy saving respect LIV_MIG (%)	50.17	52.86	63.75

Motivated by these considerations, goal of the last set of numerical results reported in this subsection is to give some insight about the actual energy consumptions of the proposed SCBM when it runs atop EWTCP and newReno SPTCP, respectively. For this purpose, a number of application scenarios featured by: (i) different sizes of the migrated VMs; (ii) migration downtimes; and, (iii) different values of the ratio $\left(\overline{w}/\widehat{R}\right)$ have been considered. In order to bypass the possible bias in the attained results induced by the reduced transmission bandwidth of the 3G WNIC (see the 3-rd row of Table 8), the equal-balanced EWTCP connections simulated in this subsection utilize only 4G and WiFI WNICs. As a matter of this fact, the performance comparisons are carried out with respect to the newReno SPTCP that utilizes *only* the 4G and *only* the WiFi WNIC. Furthermore, in order to cover a broad spectrum of application environments, the synthetic workload of the *memtester* tool has been generated for values of the ratio $\left(\overline{w}/\widehat{R}\right)$ ranging from 0.1 to 0.4. Table 11 recaps in a synoptic way the main parameters of the simulated WiFi-SPTCP, 4G-SPTCP and WiFi/4G-EWTCP connections.

The obtained numerical results are reported in Table 12. At this regard, we begin to point out that the size of the migrated VM (resp., the tolerated downtime) increases (resp., decreases) by crossing Table 12 from the top to the bottom (see the caption of Table 12), so that harder operating conditions are considered by moving from the top to the bottom of this table. Therefore, on the basis of this remark, a comparative examination of the numerical results of Table 12 supports three main insights. First, as it could be expected, in all considered cases, the energy consumptions of the SCBM atop the WiFi-SPTCP, 4G-SPTCP and WiFi/4G-EWTCP connections increase for increasing values of the ratio: $(\overline{w}/R_{MAX})$. Second, a comparative examination of the attained energy consumptions: $\mathcal{E}_{tot}^{4G}$ and $\mathcal{E}_{tot}^{WiFi}$ of the 4G-SPTCP and WiFi-SPTCP connections confirms that the 4G transmission technology is less energy efficient than the WiFi

TABLE 11: Main parameters of the simulated WiFi-SPTCP, 4G-SPTCP and equal-balanced WiFi/4G-EWTCP connections. The reported values are compliant with the power analysis of Section IV.

Parameter	**WiFi-SPTCP**	**4G-SPTCP**	**WiFi/4G-EWTCP**
α	2	2	2
K_0 (W/(Mb/s)^2)	5×10^{-2}	5×10^{-2}	2.5×10^{-2}
R_{MAX} (Mb/s)	0.9×11	0.9×50	2×9.9
$\mathcal{P}_{setup}$ (mW)	159.46	137.83	297.29

one[9]. As a matter this fact, in the carried out tests, the per-cent energy savings: $Energy\ saving_{4G}$ of the WiFi/4G-EWTCP over the 4G-SPTCP are quite large and span the interval (see the 5-th column of Table 12): 52% – 73%, whilst the corresponding energy savings: $Energy\ saving_{WiFi}$ over the WiFi-SPTCP are somewhat more limited and range over the interval (see the last column of Table 12): 17% – 33%. Third, in all three operating cases featured by the top, middle and bottom sections of Table 12, the attained per-cent energy savings are maximum at $(\overline{w}/R_{MAX}) = 0.25$, whilst they tend to decrease at $(\overline{w}/R_{MAX}) = 0.1$ and at $(\overline{w}/R_{MAX}) = 0.4$. Intuitively, this non monotonic "bell-shaped" behavior is due to the fact that the bandwidth aggregation capability of the EWTCP is less effective when the volume of the migrated data decreases, whilst it somewhat saturates when the data to be migrated grow too much.

Overall, the final conclusion stemming from the carried out comparative tests is that the WiFi/4G-EWTCP solution is more energy efficient in supporting the proposed SCBM than the corresponding WiFi and 4G-SPTCP ones when the sizes of the migrated VMs are larger enough (we say, over 60 (Mb)) and the tolerated downtimes are strict (we say, below 250 – 200 (ms)).

X. CONCLUSION AND HINTS FOR FUTURE RESEARCH

In this paper, we developed an adaptive settable-complexity bandwidth manager (SCBM) for the constrained minimization of the network energy consumed by the live migration of VMs over wireless (possibly, mobile) 5G FOGRAN connections that utilize the MPTCP as the transport protocol. Hard constraints on both the total migration time and downtime are guaranteed by the proposed SCBM, as well as quick responsiveness to (possibly, unpredictable) fading and/or mobility-induced abrupt changes of the state of the underlying MPTCP connection. After discussing a number of related implementation aspects, we have numerically checked the implementation complexity-vs.-energy performance tradeoff attained by the proposed SCBM, as well as its energy performances under both MPTCP and SPTCP transport protocols. Finally, we have numerically compared the average energy consumptions of the SCBM against the corresponding ones of the benchmark bandwidth managers in [9], [15] under both synthetic and real-world workloads. In a nutshell, the carried out comparisons point out that the average energy savings of the proposed SCBM over the one in [9] (resp., [15]) may approach 70% (resp., 50%) under strict constraints on the tolerated downtimes.

The presented results may be extended over four main research directions.

First, in this contribution, we focus on the aggregation bandwidth-capability of the MPTCP, in order to cope with hard constraints on the migration delays. However, MPTCP may be also effectively used to cope with the failure of (some of) the underling paths by shifting the transported flows towards the still active paths. How the proposed SCBM could effectively exploit the native failure recovery capability of the MPTCP may be, indeed, a first topic for future research.

Second, since the current version of the proposed SCBM selects the dirtied memory pages to be migrated in a random way, it may happen that a same memory page is dirtied multiple times and, then, it is migrated several times during the VM migration process. In principle, multiple migrations of the dirtiest memory pages could be avoided by exploiting the information on the dirtied memory pages that is periodically provided by the hypervisor's bitmap [12]. How this information could be effectively used by the proposed SCBM is a second topic of potential interest.

Third, although the equal-balanced operating condition for the MPTCP is well motivated in the considered all-wireless FOGRAN scenario of Fig. 1, it may become more questionable when heterogeneous mixed wireless/wired scenarios are considered. This may be, for example, the case of some emerging wireless/wired network infrastructures for the implementation of the front-haul section of Fig. 1 [8]. However, the generalization of the here developed SCBM to the case of un-balanced MPTCP connections is expected to introduce two additional non-minor challenges. First, the number of migration rates to be updated scales up by a factor N equal to the number of used WNICs, so that it is expected that the implementation complexity of the resulting SCBM will also increase by the same factor. Second, under the general power formula in (14) for the unbalanced MPTCP, the resulting (log-trasformed) objective function in Eq. (45) is *no longer convex*. This means, in turn, that gradient-based solving approaches do no longer guarantee the convergence to the global minimum of the energy objective function $\mathcal{E}_{tot}$. In principle, metaheuristic-based optimization methods could be utilized, in order to escape local minima. This approach has been, indeed, recently pursued in [43] for solving

[9] Obviously, this drawback is counter-balanced by the long-range high-broadband mobile cellular coverages supported by the 4G transmission technology.

TABLE 12: Comparative energy performances of the proposed SCBM under the simulated WiFi-SPTCP, 4G-SPTCP and equal-balanced WiFi/4G-EWTCP connections at: $\widehat{R} = R_{MAX}$ and $\Delta_{TM} = 37$ (s). $Energy\ saving_{4G}$ (%) and $Energy\ saving_{WiFi}$ (%) are the attained per-cent energy savings of the WiFi/4G-EWTCP solution over the 4G-SPTCP and WiFi-SPTCP ones, respectively. **Top**: $M_0 = 64$ (Mb) and $\Delta_{DT} = 200$ (ms); **Middle**: $M_0 = 128$ (Mb) and $\Delta_{DT} = 150$ (ms); **Bottom**: $M_0 = 256$ (Mb) and $\Delta_{DT} = 75$ (ms).

$\left(\frac{\overline{w}}{R_{MAX}}\right)$	$\mathcal{E}_{tot}^{4G}$ (J)	$\mathcal{E}_{tot}^{WiFi}$ (J)	$\mathcal{E}_{tot}^{EWTCP}$ (J)	$Energy\ saving_{4G}$ (%)	$Energy\ saving_{WiFi}$ (%)
0.10	73.95	42.93	35.42	52.10	17.49
0.25	108.17	51.30	40.77	62.30	20.52
0.40	113.75	55.68	44.87	60.50	19.40
0.10	178.90	76.13	60.05	66.40	21.12
0.25	214.85	84.86	64.31	70.06	24.21
0.40	240.72	96.75	75.46	68.65	22.00
0.10	323.84	145.52	101.16	68.76	30.48
0.25	413.14	163.88	109.45	73.50	33.21
0.40	552.90	215.51	160.57	70.96	25.49

the (somewhat related) non-convex problem of the energy-efficient VM placement and server consolidation in large-scale data centers. How to apply this meta-heuristic based approaches to the case of the energy-efficient management of the migration bandwidths of un-balanced wireless MPTCP connections is an additional topic for future research.

Finally, the performance results presented in this paper rely on numerical simulations. At this regard, we note that a final goal of the on-going project: GAUChO[10] funded by the Italian MIUR is to prototype a testbed that implements the main building blocks of the 5G FOGRAN infrastructure of Fig. 1.

.

APPENDIX A DERIVATION OF THE DYNAMIC POWER-VS.-TRANSPORT RATE FORMULA FOR WIRELESS MPTCP CONNECTIONS

The MPTCP connection works in the steady-state under the Congestion Avoidance phase, so that the average sizes: $W_d(j)$, $j = 1, \ldots, N$, of the per-subflow CWNDs must not vary. Hence, the following sets of relationships must simultaneously hold:

$$(1 - Pr_{loss}(j)) \times I_c(j) = Pr_{loss}(j) \times D_c(j),\ j = 1, \ldots, N, \tag{61}$$

where $Pr_{loss}(j)$ is the loss segment probability of the j-th subflow. Eq. (61) guarantees that the ACK-induced average increment: $(1 - Pr_{loss}(j)) \times I_c(j)$ of the j-th window size $W_d(j)$ equates the corresponding loss segment-induced average decrement: $Pr_{loss}(j) \times D_c(j)$. Since $Pr_{loss}(j)$ remains, by design, largely less than the unit during the Congestion Avoidance phase[11], we have that: $(1 - Pr_{loss}(j)) \approx 1$, so that Eq. (61) may be replaced by by the following one:

$$I_c(j) = Pr_{loss}(j) \times D_c(j), \quad j = 1, \ldots, N. \tag{62}$$

Now, according to Table 1, the virtualization of the network resources performed by the 5G Network Processor of Fig. 1 guarantees, by design, the isolation of the bandwidths allocated to the migrating VMs. Hence, it is reasonable to assume that the segment loss events affecting the virtualized 5G FOGRAN of Fig. 1 are mainly due to fading and/or device mobility [4], [20], [21]. This means, in turn, that, according to the cross-layer analysis reported, for example, in [44]–[46], $Pr_{loss}(j)$ in (62) scales down as a power of the per-flow dynamic power $\mathcal{P}(j)$, so that we can write:

$$Pr_{loss}(j) = \frac{\Omega(j)}{\mathcal{P}(j)^m}. \tag{63}$$

In Eq. (63), $\Omega(j)$ depends on the transmission technology employed by the j-th WNIC and the (dimensionless) positive exponent: $m > 0$ is mainly dictated by the the statistical property of the fading and device mobility pattern. Hence, after posing: $\alpha \triangleq 1/m$ and, then, introducing (63) into (62), Eq. (14) follows from quite direct (but somewhat tedious) algebraic manipulations of the expressions of $I_c(j)$ and $D_c(j)$ in Table 4.

REFERENCES

[1] E. Baccarelli, P. G. Vinueza Naranjo, M. Scarpiniti, M. Shojafar, and J. H. Abawajy, “Fog of everything: energy efficient networking computing architectures, research challenges, and a case study,” IEEE Access, vol. 5, pp. 9882–9910, 2017.

[2] S. Barré, C. Paasch, and O. Bonaventure, “MultiPath TCP: from theory to practice,” in Proc. of International Conference on Research in Networking, 2011, pp. 444–457.

[3] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, “A survey of mobile cloud computing application models,” IEEE Communication Surveys & Tutorials, vol. 16, no. 1, pp. 393–413, 2014.

[4] E. Baccarelli, M. Biagi, R. Bruno, M. Conti, E. Gregori, “Broadband wireless access network: a roadmap on emerging trends and standards,” in Broadband services: Business models and technologies for community networks, Wiley, 2005.

[5] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, “A survey on virtual machine migration: challenges, techniques and open issues,” IEEE Communication Surveys & Tutorials, 2018.

[6] L. Gao, T. H. Luan, W. Zhou, and S. Yu, “Fog computing and its application in 5G,” in 5G Mobile Communications, W. Xiang, K. Zheng, and X. Shen, Eds. Springer, 2017, pp. 571–593.

[10] https://www.gaucho.unifi.it/

[11] Typical values of $Pr_{loss}(j)$ during the Congestion Avoidance phase are below 1% [10].

[7] N. Cordeschi, M. Shojafar, D. Amendola, E. Baccrelli, "Energy-efficient adaptive networked datacenters for the QoS support of real-time applications," The Journal of Supercomputing, vol. 71, no. 2, pp. 448–478, February 2015.

[8] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks – A technology overview," IEEE Communication Surveys & Tutorials, vol. 17, no. 1, pp. 405–426, 2015.

[9] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in Proceedings of the Second Conference on Symposium on Networked Systems Design Implementation, Boston, USA, 2005, pp. 273–286.

[10] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath TCP: Analysis, design, and implementation," IEEE/ACM Transactions on Networking, vol. 24, no. 1, pp. 596–609, February 2016.

[11] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions," Proceedings of the IEEE, vol. 102, no. 1, pp. 11–31, January 2014.

[12] D. Chisnall, The Definitive Guide to the Xen Hypervisor, 1st ed. Prentice Hall, 2007.

[13] A. Nika, Y. Zhu, N. Ding, A. Jindal, Y. C. Hu, X. Zhou, B. Y. Zhao, and H. Zheng, "Energy and performance of smartphone radio bundling in outdoor environments," in Proc. of the 24th International Conference on World Wide Web (WWW 2015), Florence, Italy, May 18–22 2015.

[14] Y.-S. Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, and R. J. Gibbens, "How green is Multipath TCP for mobile devices?" in Proc. of the 4th workshop on All things cellular: operations, applications, & challenges, Chicago, Illinois, USA, August 22 2014, pp. 3–8.

[15] E. Baccarelli, D. Amendola, and N. Cordeschi, "Minimum-energy bandwidth management for QoS live migration of virtual machines," Computer Networks, vol. 93, pp. 1–22, December 2015.

[16] F. Teka, C.-H. Lung, and S. A. Ajila, "Nearby live virtual machine migration using cloudlets and multipath TCP," Journal of Cloud Computing: Advances, Systems and Applications, vol. 5, no. 12, pp. 1–21, 2016.

[17] R. Nasim and A. J. Kassler, "Network centric performance improvement for live VM migration," in Proc. of IEEEE 8th International Conference on Cloud Computing, New York, NY, USA, June 27 – July 2 2015.

[18] Y. Qiu, C.-H. Lung, S. Ajila, and P. Srivastava, "LXC container migration in Cloudlets under Multipath TCP," in Proc. of IEEE 41st Annual Computer Software and Application conference, Turin, Italy, July 4–8 2017.

[19] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath TCP," in Proc. of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design (CellNet 2012), Helsinki, Finland, August 13 2012.

[20] Y.-C. Chen, Y.-S. Lim, Y.-S. Gibbens, R. J. Nahum, E. M. Khalili, and D. Towsley, "A measurement-based study of multipath TCP performance over wireless networks," in Proc. of the 2013 Conference on Internet Measurement conference (IMC 2013), Barcelona, Spain, October 23–25 2013, pp. 455–468.

[21] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, "WiFi, LTE, or both? Measuring multi-homed wireless interface performance," in Proc. of the 2014 Conference on Internet Measurement Conference (IMC 2014), Vancouver, BC, Canada, November 5–7 2014, pp. 181–194.

[22] E. Cuervo and A. Balasubramanian, "MAUI: making smartphone last longer with code offloading," in Proc. of 8th International Conference on Mobile Systems and Applications, San Francisco, California, USA, June 15–18 2010, pp. 49–62.

[23] B. Chun, S. Ihm, and P. Maniatis, "Clonecloud: elastic execution between mobile devices and cloud," in Proc. of 6th Conference on Computer Systems, Salzburg, Austria, April 10–13 2011, pp. 301–314.

[24] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," Journal of Network and Computer Applications, vol. 59, pp. 46–54, January 2016.

[25] M. Satyanarayanan, P. Bahl, M. Caceres, and N. Davies, "The case of VM-based cloudlets in mobile computing," IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, 2009.

[26] Y. Wu and L. Ying, "A cloudlet-based multi-lateral resource exchange framework for mobile users," in 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, April 26 – May 1 2015, pp. 927–935.

[27] Y. Zhang, D. Niyato, and W. Ping, "Offloading in mobile cloudlet systems with intermittent connectivity," IEEE Transactions on Mobile Computing, vol. 14, no. 12, pp. 2516–2529, December 2015.

[28] Z. Sanei, S. Abolfazli, A. Gani, and M. Shiraz, "SAMI: service-based arbitrated multi-tier infrastructure for mobile cloud computing," in Proc. of 1st IEEE International Conference on Services Computing (SCC), Bejing, China, August 15–17 2012, pp. 14–19.

[29] B. Zhao, Z. Xu, C. Chi, S. Zhu, and G. Cao, "Mirroring smartphones for good: a feasibility study," in Proc. of Mobile Ubiquitous Systems, Computing and Networking, 2012, pp. 26–38.

[30] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda, "Multipath congestion control for shared bottleneck," in Proc. of PFLDNeT Workshop, 2009.

[31] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley, "Overlay TCP for multi-path routing and congestion control," in Proc. of IMA Workshop on Measurements and Modeling of the Internet, 2004.

[32] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," in Proc. of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI'11), Boston, MA, USA, March 2011, pp. 1–14.

[33] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in In Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp. 225–238, June 2012.

[34] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, pp. 280–293, November 2009.

[35] G. P. Perrucci, F. H. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in Proc. of IEEE 73rd IEEE Vehicular Technology Conference, pp. 1–6, May 2011.

[36] E. Baccarelli, R. Cusani, "Recursive Kalman-type optimal estimation and detection of hidden Markov chain," Signal Processing, vol. 51, no. 1, pp. 55–64, May 1996.

[37] D. Breitgand, G. Kutiel, and D. Raz, "Cost-aware live migration of services in the cloud," in Proc. of the 3rd Annual Haifa Experimental Systems Conference (SYSTOR 2010), New York, USA, May 24–26 2010, pp. 1–6.

[38] K. Z. Ibrahim, S. Hofmeyr, C. Iancu, and E. Roman, "Optimized pre-copy live migration for memory intensive applications," in Proc. of 2011 International Conference for High Performance Computing Networking, Storage and Analysis, Seattle, USA, November, 12–18 2011, pp. 1–11.

[39] M. S. Bazaraa, H. D. Sherali, and C. Shetty, Nonlinear Programming. Wiley, 2006.

[40] E. Baccarelli, R. Cusani, S. Galli, "A novel adaptive receiver with enhanced channel tracking capability for TDMA-based mobile radio communications," IEEE Journal on Selected Areas in Communications (JSAC), vol. 16, no. 9, pp. 1630–1639, December 1998.

[41] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, "Cloud container technologies: a state-of-the-art review," IEEE Transactions on Cloud Computing, 2017.

[42] CRIU, "https://criu.org/Main_Page," 2016.

[43] M. Scarpiniti, E. Baccarelli, P. G. Vinueza Naranjo, and A. Uncini, "Energy performance of heuristics and meta-heuristics for real-time joint resource scaling and consolidation in virtualized networked data centers," The Journal of Supercomputing, vol. 74, no. 5, pp. 2161–2198, May 2018.

[44] E. Baccarelli, M. Biagi, "Power-allocation policy and optimized design of multiple-antenna systems with imperfect channel estimation," IEEE Transactions On Vehicular Technology, vol. 53, no. 1, pp. 136–145, January 2004.

[45] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of queuing with adaptive modulation and coding over wireless links," in Proc. of 2003 IEEE Military Communications Conference (MILCOM 2003), Boston, MA, USA, October 13–116 2003.

[46] N. Cordeschi, T. Patriarca, and E. Baccarelli, "Stochastic traffic engineering for real-time applications over wireless networks," Journal of Network and Computer Applicartions, vol. 35, no. 2, pp. 681–694, March 2012.

ENZO BACCARELLI He is Full Professor in Information and Communication Engineering at the DIET Department of Sapienza University of Rome. He received the Laurea degree in Electronic Engineering and the Ph.D. degree in Communication Theory and Systems, both from Sapienza University of Rome. In 1995, he received the Post-Doctorate degree in Information Theory and Applications from the INFOCOM Department of Sapienza University of Rome. His current research focuses on data networks, distributed computing, networked data centers and Fog computing. From 2005 to 2010 he served as Associate Editor for IEEE Communications Letters. He was the national coordinator of several MIUR projects.

MICHELE SCARPINITI He received the Laurea degree in Electrical Engineering with honors from the Sapienza University of Rome, Italy, in 2005 and the Ph.D. in Information and Communication Engineering in 2009. From March 2008 he is an Assistant Professor of Circuit Theory and Multimedia Signal Processing with the Department of Information Engineering, Electronics and Telecommunications, at Sapienza University of Rome. His present research interests include nonlinear adaptive filters, audio processing and neural networks for signal processing, ICA and blind signal processing. He is a member of the Intelligent Signal Processing and MultiMedia (ISPAMM) laboratory, an interdisciplinary research group working at the DIET department of Sapienza University of Rome. The ISPAMM research aims at the design and development of innovative methodologies for multimedia processing. He is a senior member of IEEE, a member of the “Audio Engineering Society” (AES) and a member of the “Società Italiana Reti Neuroniche” (SIREN). He is also in the board of the AES Italian Section.

ALIREZA MOMENZADEH He is a researcher at the DIET Department of Sapienza University of Rome. He received a Bachelor of Civil Engineering from Estahban University in Iran and a Master degree in Structural Engineering from University Technology of Malaysia in 2015. His current research activity embraces nonlinear optimization through hybrid methods, Fog computing architectures and related sensor/actuator-based control applications.